LD-DETR: Loop Decoder DEtection TRansformer for Video Moment Retrieval and Highlight Detection

Pengcheng Zhao, Zhixian He, Fuwei Zhang, Shujin Lin*, Fan Zhou

Sun Yat-Sen University, Guangzhou, China
qingchen239@gmail.com, hezhx29@mail2.sysu.edu.cn,
zhangfw5@mail2.sysu.edu.cn, linshjin@mail.sysu.edu.cn, isszf@mail.sysu.edu.cn

Abstract

Video Moment Retrieval and Highlight Detection aim to find corresponding content in the video based on a text query. Existing models usually first use contrastive learning methods to align video and text features, then fuse and extract multimodal information, and finally use a Transformer Decoder to decode multimodal information. However, existing methods face several issues: (1) Overlapping semantic information between different samples in the dataset hinders the model's multimodal aligning performance; (2) Existing models are not able to efficiently extract local features of the video; (3) The Transformer Decoder used by the existing model cannot adequately decode multimodal features. To address the above issues, we proposed the LD-DETR model for Video Moment Retrieval and Highlight Detection tasks. Specifically, We first distilled the similarity matrix into the identity matrix to mitigate the impact of overlapping semantic information. Then, we designed a method that enables convolutional layers to extract multimodal local features more efficiently. Finally, we fed the output of the Transformer Decoder back into itself to adequately decode multimodal information. We evaluated LD-DETR on four public benchmarks and conducted extensive experiments to demonstrate the superiority and effectiveness of our approach. Our model outperforms the State-Of-The-Art models on QVHighlight, Charades-STA and TACoS datasets. Our code is available at https://github.com/qingchen239/ld-detr.

1 Introduction

Video Moment Retrieval aims to identify specific moments within a video that correspond to a given text query (Liu et al. 2015; Anne Hendricks et al. 2017; Gao et al. 2017; Liu et al. 2018; Escorcia et al. 2019b). Highlight Detection evaluates the degree of relevance of different time clips to the text (Yao, Mei, and Rui 2016; Zhang et al. 2016; Gygli, Song, and Cao 2016; Yu et al. 2018; Xiong et al. 2019). With the development of digital devices and platforms, users' demand for video content has increased significantly. Finding interesting clips in videos quickly and accurately has become an important requirement. Therefore, the research on Video Moment Retrieval and Highlight Detection has attracted widespread attention.

Existing models on Video Moment Retrieval and Highlight Detection often use contrastive learning to align video and text features (Moon et al. 2023b; Sun et al. 2024; Moon et al. 2023a; Liu et al. 2024b; Xiao et al. 2024), use attention mechanism to fuse and extract multimodal information (Liu et al. 2022b; Moon et al. 2023b; Sun et al. 2024), and use a Transformer Decoder with a zero matrix as $query$ to decode the fused multimodal information (Zheng et al. 2020; Lei, Berg, and Bansal 2021).

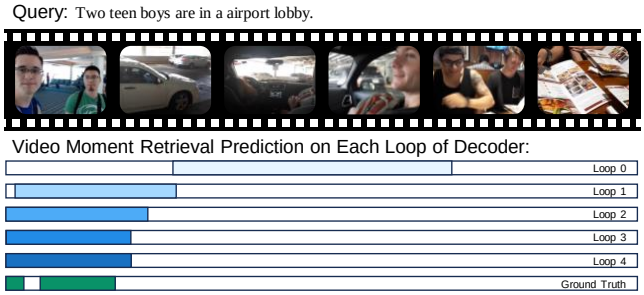

Figure 1: Loop Decoder makes Video Moment Retrieval more accurate. We visualize the Video Moment Retrieval results corresponding to the output of Loop Decoder at each loop. As the number of loops increases, the prediction gets closer and closer to the ground truth. This experiment uses M-DETR (Lei, Berg, and Bansal 2021) as the baseline.

However, the existing methods face several issues: (1) In contrastive learning, methods generally regard features from different samples as negative samples (Sun et al. 2024; Moon et al. 2023a), but the same semantic information inevitably appears in different samples (*e.g.*, "man eats" and "man drinks" both have the same information "man") (Jung et al. 2023). Treating them simply as completely negative samples hinders the performance of multimodal alignment. (2) The corresponding contents are usually small parts of the video and have strong local correlations, but the current models ignore extracting local features of the video. An intuitive approach is to use convolutional layers (LeCun et al. 1998; Krizhevsky, Sutskever, and Hinton 2012) to extract local features (Xiao et al. 2024; Li 2024), but simply stacking convolutional layers does not improve the performance of existing models. (3) Researches have shown that the Transformer Decoder inadequately processes the retrieval (Yang et al. 2024; Liu et al. 2022a; Zhu et al. 2020; Gao et al.

*Corresponding author.

2021b; Meng et al. 2021; Wang et al. 2022a; Yao et al. 2021). A bigger decoder might improve the model's retrieval abilities but also risks overfitting.

To address these issues, we proposed the Loop Decoder DEtection TRansformer model (LD-DETR): (1) Different from other methods that used an identity matrix as the target of contrastive learning, we distilled a matrix representing the correlation between samples into the identity matrix to mitigate the impact of overlapping semantic information. (2) Convolutional layers have small receptive fields. This feature allows the network to capture local information. We designed a method that enables stacked convolutional layers to extract multimodal local features more efficiently. (3) Research shows that when the Transformer Decoder's $query$ carries target information, the decoder can better decode the input information (Liu et al. 2022a), and the output of the Transformer Decoder carries exactly the target information. Inspired by this, we fed the Transformer Decoder's output back into itself as the $query$, as shown in Figure 1, to enhance its ability to decode multimodal fusion information adequately without increasing the risk of overfitting.

We evaluated LD-DETR on four public benchmarks and conducted extensive experiments to demonstrate the superiority and effectiveness of our approach.

The main contributions of our work are summarized below:

- We proposed a plug-and-play method, *Distill Align*, which takes the impact of overlapping semantic information between training samples into account when aligning multimodal features to improve the performance of the model.
- We introduced *Convolutional Fuser* to better extract local features in videos and achieved excellent results.
- We proposed a plug-and-play method, *Loop Decoder*, which improves the decoder's ability to decode multimodal fusion information adequately without causing overfitting.
- Based on the above methods, we designed a model LD-DETR for Video Moment Retrieval and Highlight Detection tasks, and verified its advancedness and effectiveness on multiple datasets. Our model outperforms the state-of-the-art models on QVHighlight, Charades-STA and TACoS datasets.

2 Related Work

Since the QVHighlight dataset (Lei, Berg, and Bansal 2021) was proposed, Video Moment Retrieval and Highlight Detection tasks have been jointly studied, and many models based on DEtection TRansformer (DETR) (Zheng et al. 2020) have been proposed. These methods mainly improve the model from three perspectives: Aligning Multimodal Features, Fusing and Extracting Multimodal Features, and Decoding Multimodal Information.

2.1 Contrastive Learning and Aligning Multimodal Features

Contrastive learning is a machine learning method that compares and contrasts different samples to improve its performance in distinguishing features and recognizing patterns. CMC (Tian, Krishnan, and Isola 2020) uses contrastive learning to map different perspectives of the same image into similar semantic spaces, demonstrating the feasibility of aligning multimodal information. MoCo (He et al. 2020) obtains the features of negative samples through a momentum-updated encoder and stores them in a queue to increase the number of samples involved in contrastive learning. CLIP (Radford et al. 2021) extracts global features for each image and text, aligning them by ensuring their correlation matrices approximate the identity matrix.

Existing Video Moment Retrieval and Highlight Detection models generally use CLIP-like methods to align video and text features. TR-DETR (Sun et al. 2024) takes the average of the video and text features encoded by the unimodal encoder as the global feature of the sample, then aligns the multimodal features using a CLIP-like method before mixing the multimodal features. CG-DETR (Moon et al. 2023a) uses two different encoders to extract global features of videos and texts of positive and negative samples respectively, and aligns them separately through a CLIP-like method. In these CLIP-like methods, the number of features involved in contrastive learning is limited by the batch size.

BM-DETR (Jung et al. 2023) proposes a weak alignment problem, the overlapping semantic information between different samples in the dataset reduces the model performance, and solves it at the multimodal feature fusion level. We believe that this problem may also reduce multimodal alignment, and we will solve it at the multimodal alignment level.

2.2 Fusing and Extracting Multimodal Features

Moment-DETR (Lei, Berg, and Bansal 2021) simply concats the video and text features and feeds them into the Transformer Encoder. UMT (Liu et al. 2022b) and QD-DETR (Moon et al. 2023b) propose a method in which Cross-Attentive Transformer Encoder uses text features to encode video features in order to remove information in video features that is irrelevant to the query text. CG-DETR (Moon et al. 2023a) goes a step further by concatenating noise onto text features based on the Cross-Attentive Transformer Encoder to better remove irrelevant information. TR-DETR (Sun et al. 2024) introduces Visual Feature Refinement to filter out irrelevant video information. These existing methods only focus on removing features that are not related to text in video features, while ignoring the temporal structure of the video itself, and leave the task of extracting video features to a Transformer Encoder (Vaswani et al. 2017). But the Transformer Encoder's global attention calculation (Bahdanau, Cho, and Bengio 2014) dilutes the weight of local details, causes the model to ignore the extraction of local features of the video. UVCOM (Xiao et al. 2024) proposes the Comprehensive Integration Module method, which has a convolutional layer to extract local multimodal features. CDIM (Li 2024) also proposes a Cross-Modal Convolutional Interaction method which stacks dilated convolutional layers to enhance the model's perceptive capability.

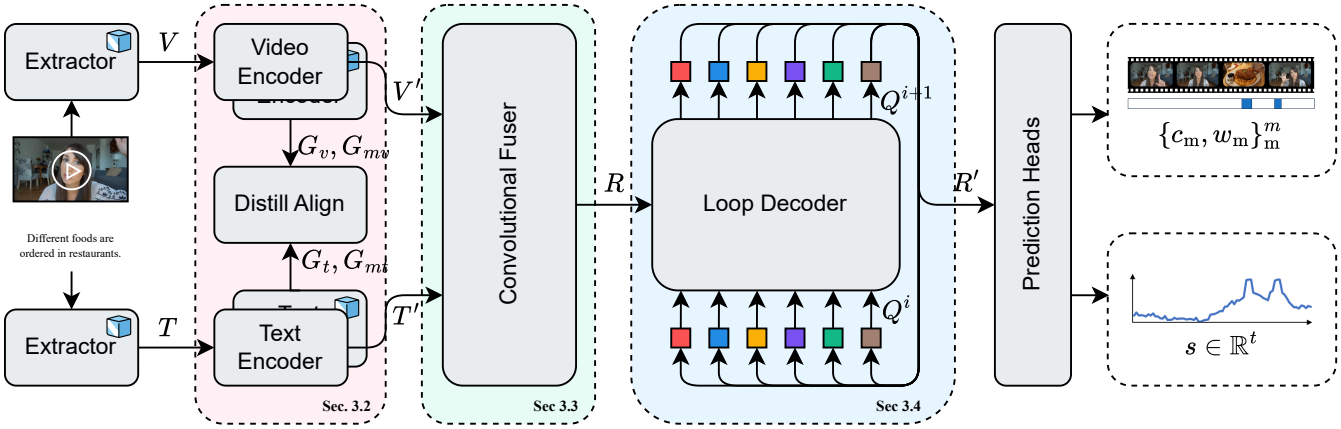

Figure 2: The overall framework of our model LD-DETR. For the methods marked with an ice cube, their parameters are not affected by gradient descent during training.

2.3 Decoding Multimodal Information

Most existing models are based on a model called DETR (Zheng et al. 2020) that is applied to object detection. Researches (Yang et al. 2024; Liu et al. 2022a; Zhu et al. 2020; Gao et al. 2021b; Meng et al. 2021; Wang et al. 2022a; Yao et al. 2021) have shown that Transformer Decoders inadequately process fused multimodal information, and propose solutions suitable for object detection tasks. MomentDiff (Li et al. 2024) notices that unevenly distributed training samples lead to a lack of generalization of the model, and introduces a diffusion model method into the decoder to solve this problem. UVCOM (Xiao et al. 2024) uses the cross-attention of text and video features as the $query$ of the Transformer Decoder. TaskWeave (Yang et al. 2024) uses a decoder following DAB-DETR (Liu et al. 2022a) to decode Video Moment Retrieval features, and uses another decoder to decode Highlight Detection features.

3 Model and Methods

3.1 Overview of LD-DETR

Given a video containing t clips, and a text query containing n tokens, the goal of Video Moment Retrieval and Highlight Detection is to find all moments $\{c_m, w_m\}_m^m$ related to the text query in the video, where c_m and w_m represent the center time and duration length of the m-th moment and m is the total number of predicted moments, and predict the saliency scores $s \in \mathbb{R}^t$ of all moments at the clip-level.

As shown in Figure 2, the LD-DETR can be divided into five parts: Unimodal Encoders, Distill Align, Convolutional Fuser, Loop Decoder, and Prediction Heads.

The input video and text are firstly fed into pre-trained feature extractors to extract video and text features $V \in \mathbb{R}^{b\times n\times d_v}, T \in \mathbb{R}^{b\times n\times d_t}$, where b is the batch size, d_v and d_t are the dimensions of the extracted video clip and text token features respectively. Then the video and text features are fed into two unimodal encoders $\mathrm{UE}_v(\cdot), \mathrm{UE}_t(\cdot)$ to be mapped into latent spaces $V' \in \mathbb{R}^{b\times t\times d}, T' \in \mathbb{R}^{b\times n\times d}$, where d is the hidden dimension of the model. We use the two unimodal encoders and two momentum unimodal encoders $\mathrm{UE}_{vm}(\cdot), \mathrm{UE}_{tm}(\cdot)$ to obtain global features. The two global features and two momentum global features $G_v, G_{mv}, G_t, G_{mt} \in \mathbb{R}^{b\times d}$ obtained by unimodal encoders are fed into a Distill Align method to ensure that the features are mapped to the same space. Then the mapped features V', T' are sent to the Convolutional Fuser to obtain multimodal features $R \in \mathbb{R}^{b\times t\times d}$. Then the multimodal features R are sent into the Loop Decoder together with a zero matrix $O \in \mathbb{R}^{b\times q\times d}$ to obtain the decoded features $R' \in \mathbb{R}^{b\times q\times d}$, where q is a hyperparameter representing the number of reference point. Finally, the decoded features are sent to the prediction heads to obtain predicted moments $\{c_m, w_m\}_m^m$ and saliency scores $s \in \mathbb{R}^t$.

3.2 Unimodal Encoders and Distill Align

We use a two-layer multilayer perceptron (Rumelhart, Hinton, and Williams 1986) as a unimodal encoder $\mathrm{UE}(\cdot)$ to map the extracted features $X \in \mathbb{R}^{b\times x\times d_x}$ to the latent space $X' \in \mathbb{R}^{b\times x\times d}$, where $X \in \{V, T\}$ and $x \in \{t, n\}$. We use the average of the features of the sample in the latent space X' in the clip or token dimension to obtain the global features $G \in \mathbb{R}^{b\times d}$ of each sample.

We use two learnable unimodal encoders $\mathrm{UE}(\cdot) \in \{\mathrm{UE}_v(\cdot), \mathrm{UE}_t(\cdot)\}$ and two momentum unimodal encoders $\mathrm{UE}_m(\cdot) \in \{\mathrm{UE}_{mv}(\cdot), \mathrm{UE}_{mt}(\cdot)\}$ to map the extracted features $V \in \mathbb{R}^{b\times x\times d_v}, T \in \mathbb{R}^{b\times x\times d_t}$ into a latent space, and obtaine four global features G_v, G_{mv}, G_t, G_{mt} respectively. Momentum unimodal encoders are updated from the corresponding unimodal encoders:

$$\mathrm{UE}^0_{m\theta} = \mathrm{UE}^0_{\theta}, \tag{1}$$

$$\mathrm{UE}^i_{m\theta} = m\mathrm{UE}^{i-1}_{m\theta} + (1-m)\mathrm{UE}^i_{\theta}, \quad \text{when } i > 0, \tag{2}$$

where $m \in [0, 1)$ is a momentum coefficient, X_θ means all parameters in the method X. Obtained the mapped features $V', V'_m, T', T'_m \in \mathbb{R}^{b\times x\times d}$, we calculate the global features for each sample $G_v, G_{mv}, G_t, G_{mt} \in \mathbb{R}^{b\times d}$. We push the momentum global features into momentum global features queues $Q_v, Q_t \in \mathbb{R}^{l\times d}$, where l represents the queue length.

Figure 3 shows the structure of the Distill Align. At this point, we have video global features $G_v \in \mathbb{R}^{b\times d}$, video and

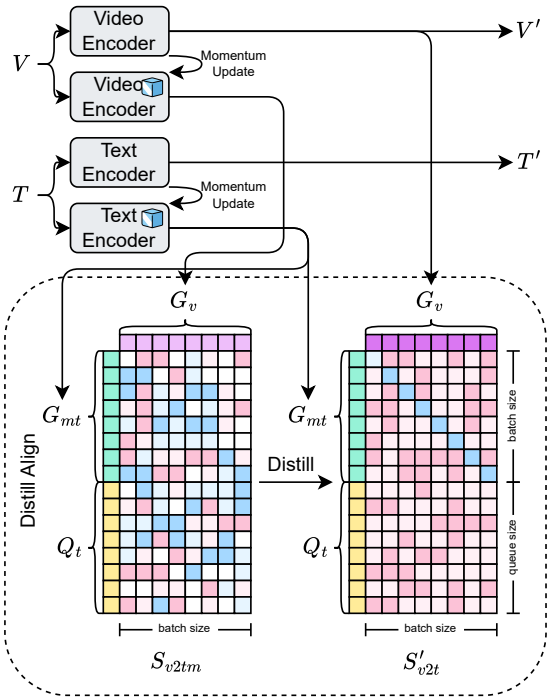

Figure 3: The structure of the Distill Align.

text momentum global features $G_{mv}, G_{mt} \in \mathbb{R}^{b\times d}$, and text momentum global features queues $Q_t \in \mathbb{R}^{l\times d}$. By calculating the cosine similarity $s(\cdot,\cdot)$, we get the video to text similarity matrix $S_{v2t}, S_{v2tm} \in \mathbb{R}^{b\times l}$ among these global features:

$$S_{v2t} = s(G_v, Q_t), \tag{3}$$
$$S_{v2tm} = s(G_{mv}, Q_t). \tag{4}$$

To mitigate the impact of overlapping semantic information, we distill the similarity matrix $S_{v2tm} \in \mathbb{R}^{b\times l}$ into an identity matrix I as the goal $S'_{v2t} \in \mathbb{R}^{b\times l}$ of multimodal alignment:

$$S'_{v2t} = \alpha S_{v2tm} + (1-\alpha)I, \tag{5}$$

where $\alpha \in [0, 1)$ is the distillation coefficient. Finally, the video-to-text alignment loss $\mathcal{L}_{v2t}$ is

$$\mathcal{L}_{v2t} = \mathrm{CE}(S_{v2t}, S'_{v2t}), \tag{6}$$

where $\mathrm{CE}(\cdot,\cdot)$ is cross entropy loss.

Similarly, we calculate the text-to-video alignment loss $\mathcal{L}_{t2v}$. The final multimodal alignment loss is

$$\mathcal{L}_{align} = (\mathcal{L}_{v2t} + \mathcal{L}_{t2v})/2. \tag{7}$$

3.3 Convolutional Fuser

Figure 4 shows the structure of the Convolutional Fuser. First, we feed the video features $V' \in \mathbb{R}^{b\times t\times d}$ and text features $T' \in \mathbb{R}^{b\times n\times d}$ mapped to the latent space into a V2T Extractor (Sun et al. 2024), a T2V Encoder (Moon et al. 2023b) and a Transformer Encoder (Vaswani et al. 2017) to obtain text-irrelevant video features $V'' \in \mathbb{R}^{b\times t\times d}$. Subsequently, the text-irrelevant video features $V'' \in \mathbb{R}^{b\times t\times d}$

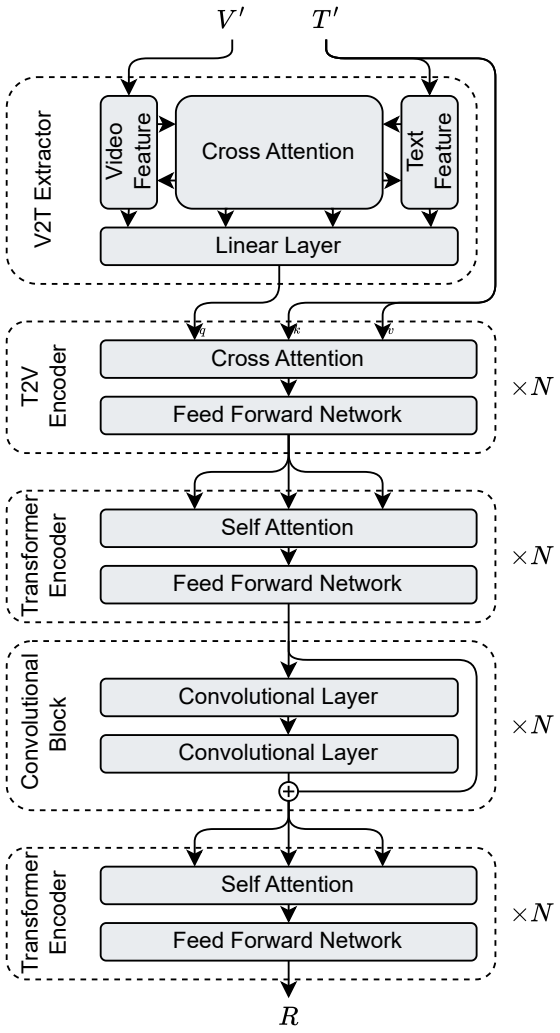

Figure 4: The structure of the Convolutional Fuser.

are fed into convolutional blocks. After passing through this residual network, we obtain local enhanced multimodal fusion features $V''' \in \mathbb{R}^{b\times t\times d}$. Finally, the locally enhanced multimodal fusion features $V''' \in \mathbb{R}^{b\times t\times d}$ are fed into another Transformer Encoder (Vaswani et al. 2017) to obtain multimodal features $R \in \mathbb{R}^{b\times t\times d}$.

V2T Extractor First, we use the video features $V' \in \mathbb{R}^{b\times t\times d}$ and text features $T' \in \mathbb{R}^{b\times n\times d}$ which have been mapped to the latent space to obtain the correlation matrix $A \in \mathbb{R}^{b\times t\times n}$ between video features and text features:

$$A_1 = \mathrm{Linear}(V'), \tag{8}$$
$$A_2 = \mathrm{Linear}(T'), \tag{9}$$
$$A_3 = \mathrm{Linear}(V')T'^T, \tag{10}$$
$$A = A_1 + A_2^T + A_3, \tag{11}$$

when matrices are added, the shorter dimension expands itself to the same length as the other matrices. Then we perform softmax on the correlation matrix A in the text and

video dimensions to obtain two other correlation matrices $A_r, A_c \in \mathbb{R}^{b\times t\times n}$. Then we get text-irrelevant video features $V'_v \in \mathbb{R}^{b\times t\times d}$:

$$T_v = A_r T', \tag{12}$$
$$V_t = A_t A_c^T V', \tag{13}$$
$$V_{cat} = [V' || T_v || V' \circ T_v || V' \circ V_t], \tag{14}$$
$$V'_{cat} = \text{Linear}(V_{cat}), \tag{15}$$
$$B = \text{Softmax}(\text{Linear}(T')), \tag{16}$$
$$T_p = T'^T B, \tag{17}$$
$$V''_{cat} = [V'_{cat} || T_p], \tag{18}$$
$$V'_v = \text{Linear}(V''_{cat}), \tag{19}$$

where $[\cdot||\cdot]$ represents concatenation, $\circ$ represents the Hadamard product.

T2V Encoder Then, we use text-irrelevant video features V'_v to obtain text-guided text-irrelevant video features $V'' \in \mathbb{R}^{b\times t\times d}$:

$$Q_v = \text{Linear}(V'_v), \tag{20}$$
$$K_t = \text{Linear}(T'), \tag{21}$$
$$V_t = \text{Linear}(T'), \tag{22}$$
$$\text{Attention}(Q_v, K_t, V_t) = \text{Softmax}\left(\frac{Q_v K_t^T}{d}\right) V_t, \tag{23}$$
$$V'' = \text{FFN}(\text{Attention}(Q_v, K_t, V_t)). \tag{24}$$

Transformer Encoder The Transformer Encoder here is no different from that in other papers (Vaswani et al. 2017). It should be noted that the two Transformer Encoders here do not share parameters.

Convolutional Blocks Subsequently, the text-irrelevant video features $V'' \in \mathbb{R}^{b\times t\times d}$ are fed into convolutional blocks. Following previous work in image recognition, we use residual blocks $\text{RB}(\cdot)$ similar to ResNet (He et al. 2016):

$$X_i = \text{RB}_i(X_{i-1}) = \sigma(X_{i-1} + \mathcal{F}(X_{i-1})), \tag{25}$$
$$\mathcal{F}(X_{i-1}) = \text{BN}(\text{Conv}(\sigma(\text{BN}(\text{Conv}(X_{i-1}))))), \tag{26}$$

where $\text{Conv}(\cdot)$ represents an Convolutional Layer, $\text{BN}(\cdot)$ represents the batch normalization, and $\sigma(\cdot)$ represents an activation function. We stack N residual blocks to extract local information in video features:

$$X_0 = V'', \tag{27}$$
$$X_i = \text{RB}_i(X_{i-1}), \qquad \text{when } i > 0, \tag{25}$$
$$V''' = X_N. \tag{28}$$

After passing through this residual network, we obtain local enhanced multimodal fusion features $V''' \in \mathbb{R}^{b\times t\times d}$.

3.4 Loop Decoder

Research shows that when the Transformer Decoder's $query$ carries target information, the decoder can better decode the input information (Liu et al. 2022a), and the output of the Transformer Decoder carries exactly the target information. Inspired by this, we feed the Transformer Decoder's output back into itself as the $query$ to enhance its ability to decode multimodal fusion information.

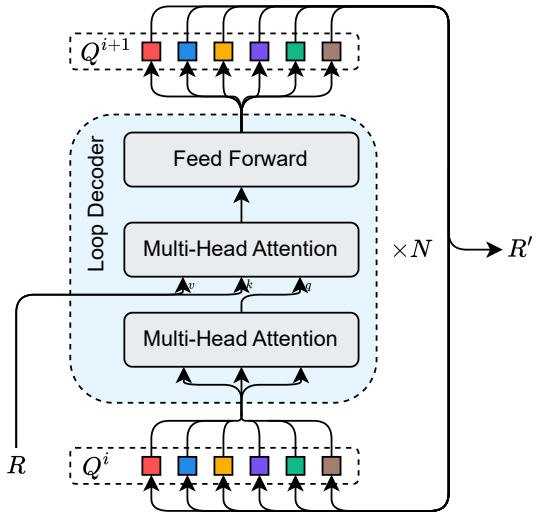

Figure 5: The structure of the Loop Decoder.

Figure 3 shows the structure of the Loop Decoder. We feed a zero matrix $O \in \mathbb{R}^{b\times q\times d}$ and multimodal features $R \in \mathbb{R}^{b\times t\times d}$ into the Transformer Decoder $\text{TD}(\cdot,\cdot)$ (Vaswani et al. 2017). Decoded features $R' \in \mathbb{R}^{b\times q\times d}$ are obtained after N loops of Transformer Decoder:

$$Q^0 = O, \tag{29}$$
$$Q^i = \text{TD}(Q^{i-1}, R), \qquad \text{when } i > 0, \tag{30}$$
$$R' = Q^N, \tag{31}$$

where q is a hyperparameter representing the number of reference points. Compared with a bigger decoder, Loop Decoder does not introduce new parameters and does not cause overfitting.

3.5 Prediction Heads

We use prediction heads similar to QD-DETR (Moon et al. 2023b) and TR-DETR (Sun et al. 2024).

Video Moment Retrieval Prediction Head We feed the decoded features $R' \in \mathbb{R}^{b\times q\times d}$ into a multi-layer perceptron to obtain q predicted moments $\{c_\text{m}, w_\text{m}\}_\text{m}^q$:

$$\{s, e\} = \text{MLP}(R'), \tag{32}$$

where s and e represents the start and end of one predicted moments. At the same time, the decoded features R and multimodal features $R' \in \mathbb{R}^{b\times t\times d}$ are used to obtain the confidence of each predicted moments $p \in \mathbb{R}^q$:

$$p = \text{MLP}(R') + \text{Sigmoid}^{-1}(\text{MLP}(R')), \tag{33}$$

where $\text{Sigmoid}^{-1}(\cdot)$ represents reverse sigmoid. In this way, we get all moments $\{c_\text{m}, w_\text{m}\}_\text{m}^m$ related to the text query in the video.

Highlight Detection Prediction Head After getting all the moments, we take out the decoded features corresponding to all the clips in the moments $R'' \in \mathbb{R}^{b \times t' \times d}$, where t' represents the amount of total clips in the Video Moment Retrieval prediction, feed them into a Gated Recurrent Unit (Chung et al. 2014), and use the hidden state as the new global features of the video $G'_v \in \mathbb{R}^{b \times d}$:

$$o = \text{GRU}(R''), \tag{34}$$

$$G'_v = \text{GRU}_\theta, \tag{35}$$

where o represents the output of the Gated Recurrent Unit which we do not care about, GRU_θ represents the Gated Recurrent Unit's hidden state. Then, we calculate the similarity between the new global features G'_v and the features V' of each clip:

$$S = G'_v V'^T \tag{36}$$

and finally get saliency scores $s \in \mathbb{R}^t$:

$$M = \text{Linear}(R' \circ S + R'), \tag{37}$$

$$s = \text{sum}(M)/d, \tag{38}$$

where $\circ$ represents the Hadamard product, $\text{sum}(\cdot)$ means to sum matrix elements over columns, d is the hidden dimension of the model.

3.6 Objective Losses

The objective loss function of LD-DETR $\mathcal{L}_{total}$ is

$$\mathcal{L}_{total} = \mathcal{L}_{mr} + \mathcal{L}_{hd} + \lambda_{aligh}\mathcal{L}_{align}, \tag{39}$$

where

$$\mathcal{L}_{mr} = \mathcal{L}_{mom} + \lambda_{CE}\text{CE}(\hat{y}, y), \tag{40}$$

$$\mathcal{L}_{mom} = \lambda_{L1}||m - \hat{m}|| + \lambda_{gIoU}\mathcal{L}_{gIoU}(m, \hat{m}), \tag{41}$$

$$\mathcal{L}_{hd} = \lambda_{marg}\mathcal{L}_{marg} + \lambda_{cont}\text{RAC}(X_r^{\text{pos}}, X_r^{\text{neg}}) \tag{42}$$

$$\mathcal{L}_{marg} = \max(0, \Delta + S(x^{\text{low}}) - S(x^{\text{high}})), \tag{43}$$

where y and $\hat{y}$ are the ground-truth of either to foreground or background and its correspond prediction, m and $\hat{m}$ are ground-truth moment and its correspond prediction, IoU loss $\mathcal{L}_{gIoU}(\cdot, \cdot)$ is from a previous work (Rezatofighi et al. 2019), Δ is the margin, $S(\cdot)$ is the saliency score estimator, x^{high} and x^{low} are video tokens from two pairs of high and low-rank clips respectively, $\text{CE}(\cdot, \cdot)$ is cross entropy loss, $\text{RAC}(\cdot, \cdot)$ is the rank-aware contrastive loss (Hoffmann et al. 2022), X_r^{pos} and X_r^{neg} are the set of samples with a higher and lower rank than the iteration index respectively.

4 Experiments

4.1 Datasets

We evaluated our model on four prevalent Video Moment Retrieval and Highlight Detection public benchmarks: QVHighlights (Lei, Berg, and Bansal 2021), CharadeSTA (Gao et al. 2017), TACoS (Regneri et al. 2013), and TVSum (Song et al. 2015).

Since the data set limits the number of submissions, in Comparison with Models, we conducted multiple experiments and gave the result of the experiment that was most likely to get the best result.

Because QVHighlight (Lei, Berg, and Bansal 2021) is the only existing data set that supports both Video Moment Retrieval and Highlight Detection. In Ablation Studies, we conducted all experiments on the QVHighlight dataset *val* split. We conducted each experiment five times, using 1, 23, 456, 7,890 and 12,345 as random seeds respectively, and the average and variance of all experimental results are given.

4.2 Metrics

We adopted the same evaluation metrics with previous works. To be specific, we computed Recall@1 with IoU threshold $\theta_{IoU} = 0.5$ and 0.7, mean average precision (mAP) with $\theta_{IoU} = 0.5$ and 0.7, and mAP with a series of thresholds $[0.5 : 0.05 : 0.95]$ for Video Moment Retrieval on QVHighlights. mAP and HIT@1 where positive samples were defined as with the saliency score of v were adopted for Highlight Detection. On Charades-STA and TACoS datasets, we utilized Recall@1 with $\theta_{IoU} = \{0.3, 0.5, 0.7\}$ and mIoU to measure the Video Moment Retrieval performance. For TVSum, mAP and Top-5 mAP are adopted, respectively.

4.3 Experimental Settings

In all experiments, we adopted CLIP (Radford et al. 2021) and Slowfast (Feichtenhofer et al. 2019) as extractors to extract video features, and adopted CLIP (Radford et al. 2021) to extract text features. In some experiments, adopted PANN (Kong et al. 2020) to extract audio features.

By default, our model trained 200 epochs with AdamW optimizer (Loshchilov 2017) using a learning rate of 1e-4, a batch size of 32, a hidden dimension of 256, a queue length of 65,536, a momentum coefficient of 0.995, a distillation coefficient of 0.4, 5 layers of Convolutional Blocks, 10 number of reference points, and 3 loops of the Loop Decoder. In Charade-STA dataset, we use 4 layers of Convolutional Blocks. In the TACoS dataset, we used a distillation coefficient of 0.7. In the Charade-STA dataset, we trained 100 epochs using a distillation coefficient of 0.3. In the TVSum dataset, we tried every method and changed every hyperparameter to achieve a better result. Because of this dataset's characteristics, the hyperparameters in each experiment are different. So, we did not record them.

4.4 Comparison with Other Models

Figure 6 visualizes comparison on LD-DETR and other models. Table 1 reports the LD-DETR's performance on joint Video Moment Retrieval and Highlight Detection tasks on QVHighlight dataset. All of the models given in the table don't use pre-training. The models in the table are divided into three categories according to the extractors used. Our model outperforms all existing models, even those using more extracted features. Table 2 reports the LD-DETR's performance on Video Moment Retrieval on TACoS dataset and Charades-STA dataset. Benefiting from our proposed method, the LD-DETR model outperforms the State-Of-The-Art models on QVHighlight, Charades-STA and TACoS datasets.

Table 3 reports the LD-DETR's performance on Highlight Detection on TV-Sum dataset. The TV-Sum data set is too

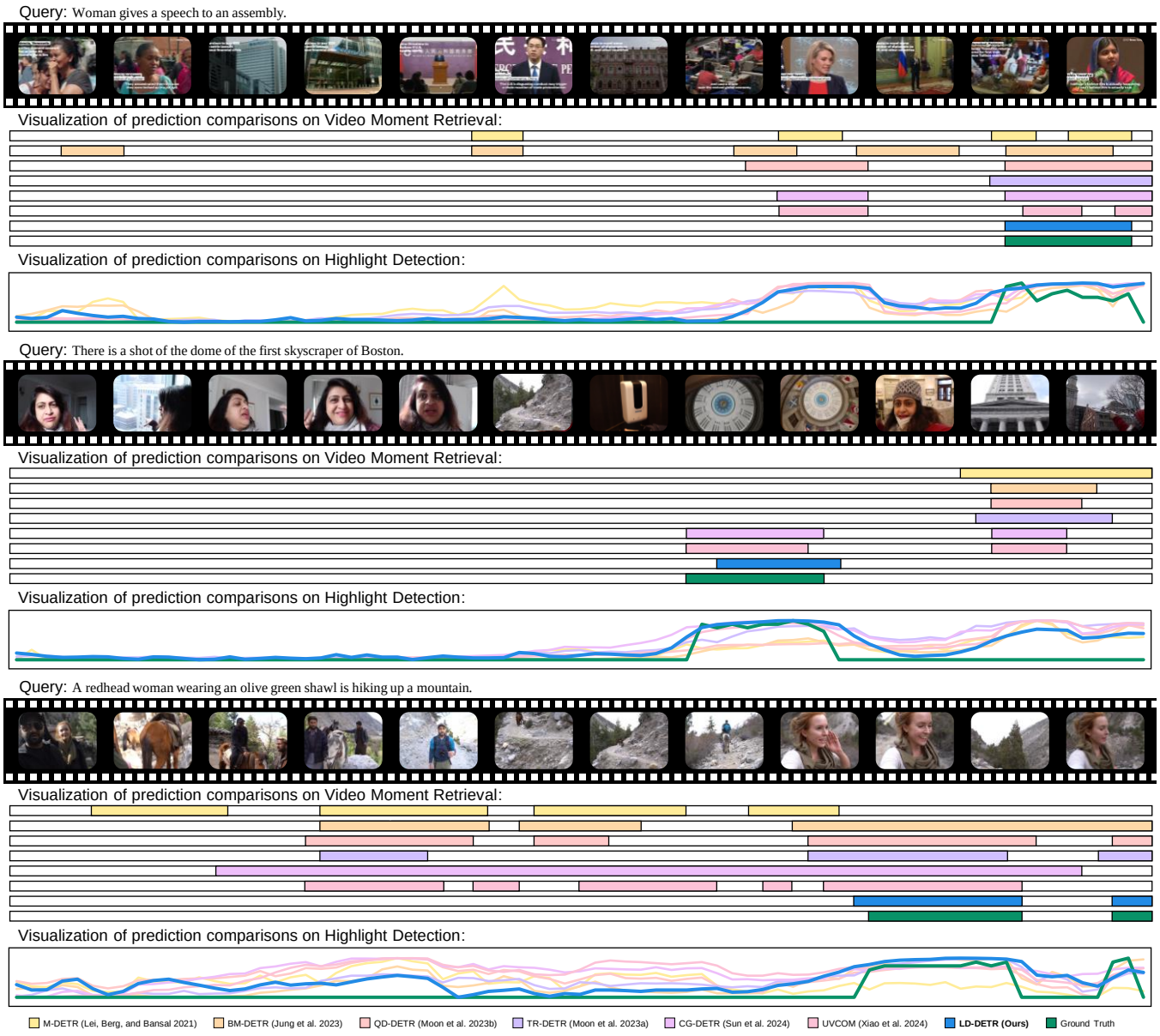

Figure 6: Visualization comparison on Video Moment Retrieval and Highlight Detection.

small, with only 4 training samples and 1 testing sample for each category. During training, the models quickly overfit. Even so, LD-DETR model still achieves not bad results on the TV-Sum dataset.

4.5 Ablation Studies

Ablation Studies on Distill Align Table 4 shows the performance of Distill Align on multiple models. It demonstrates Distill Align as a plug-and-play method to improve the performance of multiple models. Through the Distill Align method, the video and text features mapped to the latent space are aligned to the same semantic space, and the overlapping semantic information in different training samples is taken into account during the alignment.

Figure 7 and Table 5 show that the Distill Align method enables the model to achieve better results without occupying too much GPU memory. It can be seen that, when use Bigger Batch Size, as the batch size increases, the effect of the model becomes better. However, bigger batch size causes the GPU memory occupied by the model to increase, and it is more difficult for the model to converge as the batch size increases. Specifically, when batch size = 1,024, we conducted 5 experiments, and only one of them successfully converged the model. When using Distill Align, as the queue length increases, the effect of the model continues to get better, while the GPU memory occupied is much smaller.

Model	Video Moment Retrieval					Highlight Detection	
	R1		mAP			>=Very Good	
	@0.5	@0.7	@0.5	@0.75	Avg.	mAP	HIT@1
Slowfast + CLIP (5.5 GB)							
BeautyThumb (Song et al. 2016)	-	-	-	-	-	14.36	22.88
DVSE (Liu et al. 2015)	-	-	-	-	-	18.75	21.79
MCN (Anne Hendricks et al. 2017)	11.41	2.72	24.94	8.22	10.67	-	-
CAL (Escorcia et al. 2019a)	25.49	11.54	23.40	7.65	9.89	-	-
XML (Lei et al. 2020)	41.83	30.35	44.63	31.73	32.14	34.49	55.25
XML+ (Lei, Berg, and Bansal 2021)	46.69	33.46	47.89	34.67	34.90	35.38	55.06
Moment-DETR (Lei, Berg, and Bansal 2021)	52.89	33.02	54.82	29.40	30.73	35.69	55.60
Localizer (Yu et al. 2024)	54.50	36.50	-	-	32.30	-	-
UniVTG (Lin et al. 2023)	58.86	40.86	57.60	35.59	35.47	38.20	60.96
MomentDiff (Li et al. 2024)	57.42	39.66	54.02	35.73	35.95	-	-
VMRNet (Miao, Zhang, and Xu 2024)	59.94	42.84	55.56	37.75	36.87	-	-
LLaViLo (Ma et al. 2023)	59.23	41.42	9.72	-	36.94	-	-
MH-DETR (Xu et al. 2024)	60.05	42.48	60.75	38.13	38.38	38.22	60.51
QD-DETR (Moon et al. 2023b)	62.40	44.98	62.52	39.88	39.86	38.94	62.40
CDIM (Li 2024)	60.51	45.53	61.36	41.05	39.94	37.69	60.05
BM-DETR (Jung et al. 2023)	60.12	43.05	63.08	40.18	40.08	-	-
MESM (Liu et al. 2024c)	62.78	45.20	62.64	41.45	40.68	-	-
EaTR (Jang et al. 2023)	61.36	45.79	61.86	41.91	41.74	37.15	58.65
LMR (Liu et al. 2024a)	64.40	47.21	64.65	43.16	42.56	-	-
TR-DETR (Sun et al. 2024)	64.66	48.96	63.98	43.73	42.62	39.91	63.42
CG-DETR (Moon et al. 2023a)	65.43	48.38	64.51	42.77	42.86	40.33	**66.21**
CDNet (Ma et al. 2024)	67.74	49.55	63.82	42.30	42.76	39.84	66.52
UVCOM (Xiao et al. 2024)	63.55	47.47	63.37	42.67	43.18	39.74	64.20
SFABD (Huang et al. 2024)	-	-	62.38	44.39	43.79	-	-
LLMEPET (Jiang et al. 2024)	66.73	49.94	65.76	43.91	44.05	40.33	65.69
UniVTG+ (Chen et al. 2024)	66.65	**52.19**	64.37	46.68	45.18	40.18	64.77
BAM-DETR (Lee and Byun 2023)	62.71	48.64	64.57	46.33	45.36	-	-
TaskWeave (Yang et al. 2024)	64.26	50.06	65.39	46.47	45.38	39.28	63.68
LD-DETR (Ours)	**66.80**	51.04	**67.61**	**46.99**	**46.41**	**40.51**	65.11
Slowfast + CLIP + PANN (11.7 GB)							
UMT (Liu et al. 2022b)	56.23	41.18	53.38	37.01	36.12	38.18	59.99
VCSJT (Zhou et al. 2024)	59.14	42.02	55.76	37.79	36.37	38.59	62.45
MIM (Li et al. 2023)	59.99	41.50	55.85	36.84	36.45	38.96	62.39
LSJT (Wang et al. 2024)	60.51	41.50	56.33	36.70	36.66	39.13	61.22
MomentDiff (Li et al. 2024)	58.21	41.48	54.57	37.21	36.84	-	-
QD-DETR (Moon et al. 2023b)	63.06	45.10	63.04	40.10	40.19	39.04	62.87
UVCOM (Xiao et al. 2024)	63.81	48.70	64.47	44.01	43.27	39.79	64.79
LD-DETR (Ours)	**65.76**	**50.71**	**66.06**	**46.62**	**45.85**	**41.00**	**67.06**
CLIP+ (233.7 GB)							
R^2-Tuning (Liu et al. 2024b)	68.03	49.35	69.04	47.56	46.17	40.75	64.20

Table 1: Jointly Video Moment Retrieval and Highlight Detection results on QVHighlights *test* split. The table categorizes the models by the used extractors, and indicates the extractors and the extracted features sizes. The best result in each category of features on *test* split in each column is highlighted in **bold**, and the 2-nd best result in is highlighted in underline. The models shown in grey only report their results on the *val* set in the paper.

Figure 8 and Table 6 show the impact of distillation on multimodal alignment in Distill Align. Among them, when Distillation Coefficient = 0.0, it is equivalent to not enabling distillation. It can be seen that when a certain degree of distillation is used, the overlapping semantic information between different training samples is taken into account by multi-modal alignment, and the effect of the model becomes better.

Model	TACoS				Charades-STA			
	R1@0.3	R1@0.5	R1@0.7	mIoU	R1@0.3	R1@0.5	R1@0.7	mIoU
CTRL (Gao et al. 2017)	18.32	13.30	-	-	-	23.63	8.89	-
ABLR (Yuan, Mei, and Zhu 2019)	19.50	9.40	-	13.40	-	-	-	-
SM-RL (Wang, Huang, and Wang 2019)	-	20.25	15.95	-	-	24.36	11.17	-
TGN (Chen et al. 2018)	21.77	18.90	-	-	-	-	-	-
SAP (Chen and Jiang 2019)	-	18.24	-	-	-	27.42	13.36	-
MIM† (Li et al. 2023)	-	-	-	-	-	43.92	25.89	-
MAN (Zhang et al. 2019)	-	-	-	-	-	46.53	22.72	-
FMAN (Jiang et al. 2023)	-	-	-	-	-	51.40	25.05	38.23
LSJT† (Wang et al. 2024)	-	-	-	-	-	44.62	25.13	-
DRN (Zeng et al. 2020)	-	23.17	-	-	-	45.40	26.40	-
UMT† (Liu et al. 2022b)	-	-	-	-	-	48.31	29.25	-
VCSJT† (Zhou et al. 2024)	-	-	-	-	-	51.21	30.22	-
SFABD (Huang et al. 2024)	-	-	-	-	-	-	30.51	-
BPNet (Xiao et al. 2021)	25.96	20.96	14.08	19.53	65.48	50.75	31.64	46.34
M-DETR (Lei, Berg, and Bansal 2021)	-	-	-	-	-	53.63	31.37	-
SCDM (Yuan et al. 2019b)	26.11	21.17	-	-	-	54.44	31.37	-
DCL (Nan et al. 2021)	38.84	29.07	19.05	28.26	67.63	50.24	32.88	48.02
HUAL (Ji et al. 2023)	-	-	-	-	70.40	52.69	28.90	48.11
VSLNet (Zhang et al. 2020a)	29.61	24.27	20.03	24.11	70.46	54.19	35.22	50.02
2D-TAN (Zhang et al. 2020b)	37.29	25.32	-	-	-	39.70	23.31	-
MMN (Wang et al. 2022b)	39.24	26.17	-	-	-	47.31	27.28	-
CrossGraphAlign (Chen et al. 2020)	39.80	26.40	-	-	-	-	-	-
CBLN (Liu et al. 2021)	38.98	27.65	-	-	-	61.13	38.22	-
CPNet (Li, Guo, and Wang 2021)	42.61	28.29	-	28.69	-	60.27	38.74	-
FVMR (Gao et al. 2021b)	41.48	29.12	-	-	-	59.46	35.48	-
SimVTP (Ma et al. 2022)	43.10	30.30	-	-	-	44.70	26.30	-
RaNet (Gao et al. 2021a)	43.34	33.54	-	-	-	60.40	39.65	-
MomentDiff (Li et al. 2024)	44.78	33.68	-	-	-	55.57	32.42	-
LLaViLo (Ma et al. 2023)	-	-	-	-	-	55.72	33.43	-
TaskWeave (Yang et al. 2024)	-	-	-	-	-	56.51	33.66	-
QD-DETR† (Moon et al. 2023b)	-	-	-	-	-	55.51	34.17	-
LMR (Liu et al. 2024a)	-	-	-	-	-	55.91	35.19	-
VLG-Net (Soldan et al. 2021)	45.46	34.19	-	-	-	-	-	-
GVL (Wang et al. 2023)	45.92	34.57	-	32.48	-	-	-	-
TR-DETR (Sun et al. 2024)	-	-	-	-	-	57.61	33.52	-
UniVTG (Lin et al. 2023)	51.44	34.97	17.35	33.60	70.81	58.01	35.65	50.10
UniVTG+ (Chen et al. 2024)	-	-	-	-	68.06	57.18	36.05	-
BM-DETR (Jung et al. 2023)	50.31	35.42	-	-	-	59.48	38.33	-
SFEN (Yang et al. 2022)	47.30	36.10	-	-	-	-	-	-
MATN (Zhang et al. 2021)	48.79	37.57	-	-	-	-	-	-
This model doesn't have a name. (Panta et al. 2024)	49.77	37.99	-	-	-	-	-	-
CDNet (Ma et al. 2024)	54.11	35.35	20.34	33.76	71.25	58.09	36.53	-
MS-DETR (Jing et al. 2023)	47.66	37.36	25.81	35.09	68.68	57.72	37.40	50.12
CG-DETR (Moon et al. 2023a)	52.23	39.61	22.23	36.48	70.43	58.44	36.34	50.13
UVCOM (Xiao et al. 2024)	-	36.39	23.32	-	-	59.25	36.64	-
R^2-Tuning (Liu et al. 2024b)	49.71	38.72	25.12	35.92	70.91	59.78	37.02	50.86
LLMEPET (Jiang et al. 2024)	52.73	-	22.78	36.55	70.91	-	36.49	50.25
UnLoc (Yan et al. 2023)	-	-	-	-	-	60.80	38.40	-
MESM (Liu et al. 2024c)	52.69	39.52	-	36.94	-	61.24	38.04	-
MCMN (Han et al. 2023)	50.24	36.78	-	-	-	**62.69**	41.38	-
BAM-DETR (Lee and Byun 2023)	56.69	41.54	**26.77**	39.31	72.93	59.95	39.38	52.33
LD-DETR (Ours)	**57.61**	**44.31**	26.24	**40.30**	**73.06**	62.28	**42.23**	**53.14**

Table 2: Video Moment Retrieval results on TACoS and Charades-STA. The best result in each column is highlighted in **bold**, and the 2-nd best result in is highlighted in underline. † denotes training with audio modality.

Model	VT	VU	GA	MS	PK	PR	FM	BK	BT	DS	Avg.
sLSTM (Zhang et al. 2016)	41.1	46.2	46.3	47.7	44.8	46.1	45.2	40.6	47.1	45.5	45.1
SG (Yuan et al. 2019a)	42.3	47.2	47.5	48.9	45.6	47.3	46.4	41.7	48.3	46.6	46.2
VESD (Cai et al. 2018)	44.7	49.3	49.6	50.3	47.8	48.5	48.7	44.1	49.2	48.8	48.1
LIM-S (Xiong et al. 2019)	55.9	42.9	61.2	54.0	60.3	47.5	43.2	66.3	69.1	62.6	56.3
Trailer (Wang et al. 2020)	61.3	54.6	65.7	60.8	59.1	70.1	58.2	64.7	65.6	68.1	62.8
MINI-Net† (Hong et al. 2020)	80.6	68.3	78.2	81.8	78.1	65.8	75.8	75.0	80.2	65.5	73.2
SL-Module (Xu et al. 2021)	86.5	68.7	74.9	86.2	79.0	63.2	58.9	72.6	78.9	64.0	73.3
SA (Badamdorj et al. 2021)	83.4	64.7	84.4	86.5	70.3	67.5	66.9	68.1	95.0	60.8	74.8
SA$^+$† (Badamdorj et al. 2021)	83.7	57.3	78.5	86.1	80.1	69.2	70.0	73.0	97.4	67.5	76.3
Joint-VA† (Badamdorj et al. 2021)	83.7	57.3	78.5	86.1	80.1	69.2	70.0	73.0	97.4	67.5	76.3
TCG† (Ye et al. 2021)	85.0	71.4	81.9	78.6	80.2	75.5	71.6	77.3	78.6	68.1	76.8
PLD-VHD (Wei et al. 2022)	84.5	80.9	70.3	72.5	76.4	87.2	71.9	74.0	74.4	79.1	77.1
UniVTG (Lin et al. 2023)	83.9	85.1	89.0	80.1	84.6	87.0	70.9	91.7	73.5	69.3	81.0
MH-DETR (Xu et al. 2024)	86.1	79.4	84.3	85.8	81.2	83.9	74.3	82.7	86.5	71.6	81.6
MIM† (Li et al. 2023)	84.4	85.8	91.3	73.9	83.1	87.1	80.1	78.2	80.3	79.6	82.4
UMT† (Liu et al. 2022b)	87.5	81.5	88.2	78.8	81.4	87.0	76.0	86.9	84.4	79.6	83.1
VCSJT† (Zhou et al. 2024)	87.5	80.7	88.6	76.6	83.6	91.0	77.6	93.3	88.9	80.0	84.8
QD-DETR (Moon et al. 2023b)	88.2	87.4	85.6	85.0	85.8	86.9	76.4	91.3	89.2	73.7	85.0
LD-DETR (Ours)	83.1	90.3	91.5	82.5	87.7	83.5	79.0	90.4	86.1	77.3	85.1
UVCOM (Xiao et al. 2024)	87.6	91.6	91.4	86.7	86.9	86.9	76.9	92.3	87.4	75.6	86.3
QD-DETR† (Moon et al. 2023b)	87.6	91.7	90.2	88.3	84.1	88.3	78.7	91.2	87.8	77.7	86.6
CG-DETR (Moon et al. 2023a)	86.9	88.8	94.8	87.7	86.7	89.6	74.8	93.3	89.2	75.9	86.8
TR-DETR† (Sun et al. 2024)	89.3	93.0	94.3	85.1	88.0	88.6	80.4	91.3	89.5	81.6	88.1
TaskWeave (Yang et al. 2024)	88.2	90.8	93.3	87.5	87.0	82.0	80.9	92.9	89.5	81.2	87.3
LLMEPET (Jiang et al. 2024)	90.8	92.0	93.8	81.5	87.5	86.0	79.6	96.2	88.0	79.0	87.4
TR-DETR (Sun et al. 2024)	89.3	93.0	94.3	85.1	88.0	88.6	80.4	91.3	89.5	81.6	88.1
CDIM (Li 2024)	85.5	95.8	90.3	90.0	88.4	88.1	79.2	97.1	88.0	80.5	88.3

Table 3: Highlight Detection results on TV-Sum. † denotes training with audio modality.

Setting	Video Moment Retrieval					Highlight Detection	
	R1		mAP			>=Very Good	
	@0.5	@0.7	@0.5	@0.75	Avg.	mAP	HIT@1
M-DETR (Lei, Berg, and Bansal 2021)	$53.63_{\pm 1.59}$	$35.78_{\pm 1.25}$	$54.99_{\pm 1.33}$	$31.28_{\pm 1.07}$	$32.09_{\pm 1.01}$	$36.41_{\pm 0.40}$	$56.92_{\pm 0.98}$
M-DETR + **Distill Align**	$\mathbf{55.56}_{\pm \mathbf{1.01}}$	$\mathbf{36.72}_{\pm \mathbf{1.43}}$	$\mathbf{56.42}_{\pm \mathbf{0.76}}$	$\mathbf{31.43}_{\pm \mathbf{0.99}}$	$\mathbf{32.66}_{\pm \mathbf{0.73}}$	$\mathbf{37.16}_{\pm \mathbf{0.36}}$	$\mathbf{58.62}_{\pm \mathbf{1.19}}$
BM-DETR (Jung et al. 2023)	$60.90_{\pm 0.79}$	$44.10_{\pm 1.43}$	$60.91_{\pm 0.96}$	$39.03_{\pm 1.46}$	$\mathbf{38.93}_{\pm \mathbf{0.98}}$	-	-
BM-DETR + **Distill Align**	$\mathbf{61.04}_{\pm \mathbf{1.02}}$	$\mathbf{44.23}_{\pm \mathbf{1.04}}$	$\mathbf{61.53}_{\pm \mathbf{0.98}}$	$\mathbf{39.12}_{\pm \mathbf{1.45}}$	$38.68_{\pm 1.09}$	-	-
QD-DETR (Moon et al. 2023b)	$61.98_{\pm 0.55}$	$47.30_{\pm 0.69}$	$62.03_{\pm 0.43}$	$41.96_{\pm 0.66}$	$41.42_{\pm 0.28}$	$38.92_{\pm 0.30}$	$62.05_{\pm 0.86}$
QD-DETR + **Distill Align**	$\mathbf{64.10}_{\pm \mathbf{0.42}}$	$\mathbf{48.53}_{\pm \mathbf{0.70}}$	$\mathbf{63.71}_{\pm \mathbf{0.37}}$	$\mathbf{43.55}_{\pm \mathbf{0.59}}$	$\mathbf{42.80}_{\pm \mathbf{0.46}}$	$\mathbf{39.58}_{\pm \mathbf{0.40}}$	$\mathbf{63.21}_{\pm \mathbf{1.05}}$
CG-DETR (Moon et al. 2023a)	$65.92_{\pm 0.22}$	$50.44_{\pm 0.54}$	$65.50_{\pm 0.22}$	$45.62_{\pm 0.68}$	$44.76_{\pm 0.26}$	$40.34_{\pm 0.20}$	$65.12_{\pm 0.64}$
CG-DETR + **Distill Align**	$\mathbf{66.11}_{\pm \mathbf{0.79}}$	$\mathbf{51.18}_{\pm \mathbf{0.89}}$	$\mathbf{65.65}_{\pm \mathbf{0.61}}$	$\mathbf{46.12}_{\pm \mathbf{0.74}}$	$\mathbf{45.23}_{\pm \mathbf{0.61}}$	$\mathbf{40.50}_{\pm \mathbf{0.23}}$	$\mathbf{65.92}_{\pm \mathbf{1.04}}$
TR-DETR (Sun et al. 2024)	$66.56_{\pm 1.06}$	$50.13_{\pm 0.89}$	$65.70_{\pm 0.79}$	$45.10_{\pm 0.78}$	$44.33_{\pm 0.51}$	$40.88_{\pm 0.19}$	$65.54_{\pm 0.45}$
TR-DETR − LGMA	$62.28_{\pm 1.08}$	$46.99_{\pm 1.21}$	$62.38_{\pm 0.72}$	$62.16_{\pm 1.41}$	$41.56_{\pm 1.12}$	$39.16_{\pm 0.32}$	$62.13_{\pm 0.46}$
TR-DETR + **Distill Align**	$66.31_{\pm 0.60}$	$49.92_{\pm 0.73}$	$65.33_{\pm 0.74}$	$44.14_{\pm 0.81}$	$43.60_{\pm 0.94}$	$\mathbf{40.89}_{\pm \mathbf{0.26}}$	$\mathbf{65.79}_{\pm \mathbf{1.02}}$
TR-DETR − LGMA + **Distill Align**	$\mathbf{67.14}_{\pm \mathbf{0.43}}$	$\mathbf{51.17}_{\pm \mathbf{0.37}}$	$\mathbf{66.21}_{\pm \mathbf{0.24}}$	$\mathbf{45.57}_{\pm \mathbf{0.42}}$	$\mathbf{44.89}_{\pm \mathbf{0.32}}$	$40.77_{\pm 0.31}$	$65.33_{\pm 0.71}$
UVCOM (Xiao et al. 2024)	$64.36_{\pm 0.44}$	$50.21_{\pm 1.06}$	$63.99_{\pm 0.27}$	$45.52_{\pm 0.68}$	$44.77_{\pm 0.53}$	$39.85_{\pm 0.21}$	$63.82_{\pm 1.05}$
UVCOM + **Distill Align**	$\mathbf{66.39}_{\pm \mathbf{0.37}}$	$\mathbf{51.54}_{\pm \mathbf{0.22}}$	$\mathbf{65.24}_{\pm \mathbf{0.49}}$	$\mathbf{46.18}_{\pm \mathbf{0.38}}$	$\mathbf{45.39}_{\pm \mathbf{0.29}}$	$\mathbf{40.72}_{\pm \mathbf{0.14}}$	$\mathbf{65.20}_{\pm \mathbf{0.73}}$

Table 4: Distill Align as a plug-and-play method can help multiple models achieve better results. LGMA is the abbreviation of Local-Global Multi-Modal Alignment which is a method used in the TR-DETR (Sun et al. 2024) model. The best result on each baseline in each column is highlighted in **bold**.

Batch Size / Queue Length	GPU Memory	Video Moment Retrieval					Highlight Detection	
		R1		mAP			>=Very Good	
		@0.5	@0.7	@0.5	@0.75	Avg.	mAP	HIT@1
Bigger Batch Size								
bs = 32, ql = 0	1,350	$66.19_{\pm 1.41}$	$49.74_{\pm 1.70}$	$65.27_{\pm 1.11}$	$44.12_{\pm 1.41}$	$43.81_{\pm 1.19}$	$40.72_{\pm 0.17}$	$65.19_{\pm 1.23}$
bs = 64, ql = 0	2,140	$66.89_{\pm 0.75}$	$50.07_{\pm 0.82}$	$65.73_{\pm 0.39}$	$44.53_{\pm 0.54}$	$43.98_{\pm 0.50}$	$40.99_{\pm 0.29}$	$65.86_{\pm 0.84}$
bs = 128, ql = 0	3,328	$66.81_{\pm 0.70}$	$50.93_{\pm 0.46}$	$65.93_{\pm 0.46}$	$45.00_{\pm 0.43}$	$44.53_{\pm 0.33}$	$40.92_{\pm 0.20}$	$65.77_{\pm 1.26}$
bs = 256, ql = 0	5,772	$66.62_{\pm 0.59}$	$49.56_{\pm 0.82}$	$65.27_{\pm 0.51}$	$43.46_{\pm 0.66}$	$43.22_{\pm 0.83}$	$40.76_{\pm 0.16}$	$65.38_{\pm 0.52}$
bs = 512, ql = 0	10,530	$65.20_{\pm 1.92}$	$47.38_{\pm 2.31}$	$63.49_{\pm 1.85}$	$41.35_{\pm 2.33}$	$40.90_{\pm 2.00}$	$40.58_{\pm 0.17}$	$64.94_{\pm 1.16}$
bs = 1,024, ql = 0	20,956	$19.15_{\pm 13.66}$	$8.70_{\pm 8.05}$	$26.29_{\pm 10.96}$	$9.45_{\pm 6.48}$	$11.73_{\pm 6.34}$	$27.13_{\pm 5.20}$	$37.18_{\pm 10.86}$
bs = 2,048, ql = 0	Out of Memory	-	-	-	-	-	-	-
Distill Align (Ours)								
bs = 32, ql = 0	1,312	$66.53_{\pm 0.96}$	$49.90_{\pm 0.31}$	$65.70_{\pm 0.89}$	$44.57_{\pm 0.48}$	$44.07_{\pm 0.13}$	$40.52_{\pm 0.24}$	$64.73_{\pm 0.95}$
bs = 32, ql = 96	1,318	$66.53_{\pm 0.32}$	$49.34_{\pm 0.33}$	$65.61_{\pm 0.85}$	$44.13_{\pm 0.35}$	$43.77_{\pm 0.49}$	$40.50_{\pm 0.16}$	$64.66_{\pm 0.24}$
bs = 32, ql = 480	1,236	$67.11_{\pm 1.49}$	$50.48_{\pm 1.03}$	$\mathbf{66.28_{\pm 1.03}}$	$45.10_{\pm 0.51}$	$44.18_{\pm 0.41}$	$40.92_{\pm 0.23}$	$65.91_{\pm 0.97}$
bs = 32, ql = 2,016	1,458	$66.76_{\pm 0.57}$	$50.48_{\pm 0.40}$	$65.73_{\pm 0.47}$	$44.83_{\pm 0.68}$	$44.22_{\pm 0.57}$	$40.81_{\pm 0.39}$	$65.09_{\pm 0.70}$
bs = 32, ql = 8,160	1,364	$66.74_{\pm 0.84}$	$\mathbf{50.81_{\pm 0.50}}$	$66.07_{\pm 0.42}$	$\mathbf{45.30_{\pm 0.52}}$	$44.53_{\pm 0.34}$	$40.76_{\pm 0.19}$	$\mathbf{65.95_{\pm 0.42}}$
bs = 32, ql = 32,736	1,620	$67.00_{\pm 0.75}$	$50.71_{\pm 1.10}$	$66.10_{\pm 0.67}$	$45.06_{\pm 0.99}$	$44.51_{\pm 0.80}$	$\mathbf{41.00_{\pm 0.21}}$	$65.64_{\pm 0.53}$
bs = 32, ql = 131,040	2,440	$\mathbf{67.14_{\pm 0.59}}$	$50.63_{\pm 0.48}$	$65.91_{\pm 0.32}$	$45.21_{\pm 0.60}$	$\mathbf{44.67_{\pm 0.30}}$	$\mathbf{41.00_{\pm 0.23}}$	$65.86_{\pm 0.82}$

Table 5: Compared with using a Bigger Batch Size, using Distill Align saves GPU memory and will get better results. This table shows the GPU memory usage and training results when using different batch sizes or queue lengths, using a Bigger Batch Size and using Distill Align. "bs" means batch size, "ql" means queue length. The experiment is conducted on an NVIDIA GeForce RTX 4090 with 24,564MiB GPU memory using TR-DETR (Sun et al. 2024) as the baseline. The best result in each column is highlighted in **bold**. This table shows the same set of experiments as Figure 7.

Distillation Coefficient	Video Moment Retrieval					Highlight Detection	
	R1		mAP			>=Very Good	
	@0.5	@0.7	@0.5	@0.75	Avg.	mAP	HIT@1
0.0	$66.28_{\pm 1.38}$	$49.91_{\pm 0.44}$	$65.64_{\pm 0.74}$	$44.60_{\pm 0.68}$	$44.23_{\pm 0.27}$	$40.81_{\pm 0.21}$	$65.03_{\pm 0.78}$
0.1	$67.20_{\pm 0.64}$	$50.50_{\pm 0.78}$	$66.20_{\pm 0.40}$	$44.83_{\pm 0.64}$	$44.50_{\pm 0.60}$	$40.85_{\pm 0.20}$	$65.58_{\pm 0.89}$
0.2	$67.15_{\pm 0.99}$	$50.50_{\pm 0.26}$	$66.28_{\pm 0.61}$	$45.24_{\pm 0.39}$	$44.63_{\pm 0.40}$	$40.98_{\pm 0.18}$	$66.00_{\pm 0.65}$
0.3	$67.07_{\pm 0.59}$	$50.62_{\pm 0.42}$	$66.19_{\pm 0.54}$	$45.29_{\pm 0.52}$	$44.61_{\pm 0.49}$	$\mathbf{41.14_{\pm 0.16}}$	$65.78_{\pm 1.26}$
0.4	$\mathbf{67.59_{\pm 0.71}}$	$\mathbf{50.89_{\pm 0.82}}$	$\mathbf{66.48_{\pm 0.37}}$	$\mathbf{45.50_{\pm 0.58}}$	$\mathbf{44.86_{\pm 0.50}}$	$41.09_{\pm 0.22}$	$\mathbf{66.32_{\pm 0.47}}$
0.5	$66.76_{\pm 0.89}$	$50.46_{\pm 0.62}$	$66.16_{\pm 0.54}$	$45.06_{\pm 0.50}$	$44.66_{\pm 0.28}$	$40.83_{\pm 0.37}$	$65.56_{\pm 0.62}$
0.6	$66.61_{\pm 0.43}$	$50.55_{\pm 0.59}$	$65.89_{\pm 0.54}$	$45.32_{\pm 0.56}$	$44.68_{\pm 0.49}$	$40.65_{\pm 0.20}$	$64.55_{\pm 0.67}$
0.7	$66.59_{\pm 0.28}$	$50.52_{\pm 0.81}$	$65.71_{\pm 0.22}$	$45.14_{\pm 0.49}$	$44.63_{\pm 0.32}$	$41.05_{\pm 0.19}$	$65.43_{\pm 0.42}$
0.8	$65.72_{\pm 0.94}$	$49.55_{\pm 1.09}$	$64.87_{\pm 0.57}$	$44.05_{\pm 1.51}$	$43.62_{\pm 1.09}$	$40.60_{\pm 0.13}$	$64.80_{\pm 0.50}$
0.9	$65.25_{\pm 0.70}$	$49.47_{\pm 0.33}$	$64.75_{\pm 0.22}$	$44.17_{\pm 0.36}$	$43.69_{\pm 0.33}$	$40.23_{\pm 0.34}$	$63.88_{\pm 1.25}$
1.0	$62.83_{\pm 0.82}$	$47.26_{\pm 0.85}$	$63.08_{\pm 0.49}$	$42.54_{\pm 0.82}$	$42.09_{\pm 0.58}$	$39.37_{\pm 0.21}$	$61.87_{\pm 0.39}$

Table 6: This table shows the impact of distillation on multimodal alignment in Distill Align. The best result on each baseline in each column is highlighted in **bold**. This table shows the same set of experiments as Figure 8.

Ablation Studies on Convolutional Fuser Table 7 shows the impact of the number of convolutional layers in Convolutional Fuser. The motion information in the video is often contained in several local clips. Convolutional layers can effectively capture the local information in the video. Compared with UVCOM (Xiao et al. 2024) (a-f), our method (g-m) can better utilize the ability of convolutional layers to extract local features. In UVCOM, as the number of convolutional layers increases, the performance of the model does not get better. But in our method, as the number of convolutional layers increases, the performance of the model becomes better. Compared with not using convolutional layers (g), just adding a convolutional layer (h) can greatly improve the model effect. As the number of convolutional layers added increases, the model is able to capture local information in the video more effectively and the effect gradually

Setting	# Conv	Video Moment Retrieval					Highlight Detection	
		R1		mAP			>=Very Good	
		@0.5	@0.7	@0.5	@0.75	Avg.	mAP	HIT@1
UVCOM (Xiao et al. 2024)								
(a)	0	$63.68_{\pm 0.43}$	$49.29_{\pm 0.86}$	$63.50_{\pm 0.46}$	$43.81_{\pm 0.44}$	$43.40_{\pm 0.41}$	$39.40_{\pm 0.13}$	$63.22_{\pm 0.46}$
(b)	1	$64.36_{\pm 0.44}$	$50.21_{\pm 1.06}$	$63.99_{\pm 0.27}$	$45.52_{\pm 0.68}$	$44.72_{\pm 0.53}$	$39.85_{\pm 0.21}$	$63.82_{\pm 1.05}$
(c)	2	$61.68_{\pm 0.59}$	$47.51_{\pm 0.54}$	$61.72_{\pm 0.74}$	$42.66_{\pm 0.53}$	$42.40_{\pm 0.47}$	$38.64_{\pm 0.43}$	$60.90_{\pm 0.81}$
(d)	4	$61.89_{\pm 0.63}$	$47.54_{\pm 0.85}$	$61.63_{\pm 0.82}$	$42.46_{\pm 0.78}$	$42.46_{\pm 0.81}$	$38.52_{\pm 0.13}$	$60.54_{\pm 0.75}$
(e)	8	$60.87_{\pm 0.23}$	$46.95_{\pm 0.63}$	$61.41_{\pm 0.32}$	$42.33_{\pm 0.50}$	$42.28_{\pm 0.23}$	$38.39_{\pm 0.23}$	$60.22_{\pm 0.92}$
(f)	16	$61.46_{\pm 0.60}$	$47.13_{\pm 0.42}$	$61.76_{\pm 0.25}$	$43.07_{\pm 0.62}$	$42.57_{\pm 0.33}$	$38.19_{\pm 0.23}$	$59.80_{\pm 0.77}$
LD-DETR (Ours)								
(g)	0	$67.07_{\pm 1.54}$	$50.96_{\pm 2.17}$	$65.79_{\pm 1.72}$	$45.17_{\pm 2.10}$	$44.18_{\pm 1.91}$	$41.42_{\pm 0.09}$	$66.30_{\pm 0.29}$
(h)	1	$68.14_{\pm 0.26}$	$52.01_{\pm 1.28}$	$66.70_{\pm 0.38}$	$46.24_{\pm 1.18}$	$45.71_{\pm 0.96}$	$41.70_{\pm 0.20}$	$\mathbf{67.41}_{\pm \mathbf{0.95}}$
(i)	2	$68.10_{\pm 0.63}$	$52.04_{\pm 0.45}$	$67.49_{\pm 0.78}$	$47.01_{\pm 0.25}$	$46.73_{\pm 0.32}$	$41.65_{\pm 0.24}$	$66.62_{\pm 0.62}$
(j)	4	$68.77_{\pm 0.40}$	$52.88_{\pm 0.66}$	$68.00_{\pm 0.38}$	$47.55_{\pm 0.66}$	$47.21_{\pm 0.46}$	$\mathbf{41.83}_{\pm \mathbf{0.15}}$	$67.30_{\pm 1.33}$
(k)	8	$68.52_{\pm 0.90}$	$52.22_{\pm 1.06}$	$67.70_{\pm 0.62}$	$67.14_{\pm 0.90}$	$47.28_{\pm 0.66}$	$41.55_{\pm 0.24}$	$66.41_{\pm 0.61}$
(l)	10	$\mathbf{69.01}_{\pm \mathbf{1.09}}$	$\mathbf{53.19}_{\pm \mathbf{0.38}}$	$\mathbf{68.43}_{\pm \mathbf{0.83}}$	$\mathbf{48.25}_{\pm \mathbf{0.59}}$	$\mathbf{47.93}_{\pm \mathbf{0.39}}$	$41.66_{\pm 0.15}$	$66.80_{\pm 0.96}$
(m)	16	$68.63_{\pm 0.97}$	$52.89_{\pm 0.62}$	$68.10_{\pm 0.64}$	$47.56_{\pm 0.36}$	$47.24_{\pm 0.46}$	$41.79_{\pm 0.15}$	$67.33_{\pm 0.42}$

Table 7: This table shows the impact of the number of convolutional layers used on model performance. "# Conv" means the number of convolutional layers. The best result in each column is highlighted in **bold**. It can be noticed that on LD-DETR, with the help of Convolutional Fuser, just adding one convolutional layer can significantly improve the model performance. As the number of convolutional layers increases, the performance of the model gets better until it reaches a limit. On UVCOM (Xiao et al. 2024), however, adding convolutional layers does not help the model to perform better.

Setting	Position	Video Moment Retrieval					Highlight Detection	
		R1		mAP			>=Very Good	
		@0.5	@0.7	@0.5	@0.75	Avg.	mAP	HIT@1
(a)	Before V2TEx	$58.72_{\pm 2.48}$	$42.43_{\pm 3.08}$	$58.45_{\pm 2.87}$	$37.44_{\pm 2.88}$	$37.14_{\pm 2.84}$	$38.20_{\pm 0.62}$	$60.39_{\pm 1.60}$
(b)	Between V2TEx & T2VEn	$66.64_{\pm 1.21}$	$49.61_{\pm 1.43}$	$65.84_{\pm 1.00}$	$43.99_{\pm 1.19}$	$43.51_{\pm 1.15}$	$41.20_{\pm 0.19}$	$66.36_{\pm 0.48}$
(c)	Between T2VEn & TrEn1	$68.87_{\pm 0.68}$	$52.96_{\pm 0.32}$	$68.07_{\pm 0.56}$	$47.54_{\pm 0.41}$	$47.51_{\pm 0.39}$	$\mathbf{41.81}_{\pm \mathbf{0.13}}$	$66.79_{\pm 0.50}$
(d)	Between TrEn1 & TrEn2	$\mathbf{69.01}_{\pm \mathbf{1.09}}$	$\mathbf{53.19}_{\pm \mathbf{0.38}}$	$\mathbf{68.43}_{\pm \mathbf{0.83}}$	$\mathbf{48.25}_{\pm \mathbf{0.59}}$	$\mathbf{47.93}_{\pm \mathbf{0.39}}$	$41.66_{\pm 0.15}$	$\mathbf{66.80}_{\pm \mathbf{0.96}}$
(e)	After TrEn2	$67.62_{\pm 0.96}$	$52.23_{\pm 0.53}$	$67.17_{\pm 0.17}$	$46.85_{\pm 0.23}$	$46.57_{\pm 0.19}$	$41.56_{\pm 0.28}$	$66.26_{\pm 0.52}$

Table 8: This table shows the performances of the model when Convolutional Blocks are placed in different locations in Convolutional Fuser. It can be noticed that the model performs best when Convolutional Blocks are placed between Transformer Encoder 1 and Transformer Encoder 2. In this table, "V2TEx" means V2T Extractor, "T2VEn" means T2V Encoder, "TrEn1" means Transformer Encoder 1, "TrEn2" means Transformer Encoder 2. The best result in each column is highlighted in **bold**.

improves until it reaches the peak (l).

Table 8 and Table 9 show ablation experiments on Convolutional Fuser. Whether you the order of the methods or remove any of them, the performance of the model will degrade.

Ablation Studies on Loop Decoder Figure 1 shows that Loop Decoder makes Video Moment Retrieval more accurate. It visualizes the Video Moment Retrieval results corresponding to the output of Loop Decoder at each loop. As the number of loops increases, the results are getting closer and closer to the ground truth. Figure 9, Table 10 and Table 11 shows the performance of Loop Decoder on multiple models. It demonstrates Loop Decoder as a plug-and-play method to improve the performance of multiple models. Through the Loop Decoder method, multimodal information is decoded more adequately. After using the Loop Decoder, the performance of the model is greatly improved. However, when using a bigger decoder of the same size, the model quickly overfits. We noticed that the effect of Loop Decoder on UVCOM (Xiao et al. 2024) is not as obvious as on other models. We speculate that this is because its Dual Branches Intra-Modality Aggregation affects the performance of our method. When we delete the Clip-Text Alignment method, the performance becomes better when the number of loops increases. But when we delete the Slot Attention method, the performance of the model becomes even better.

Setting	V2TEx	T2VEn	TrEn1	ConBl	TrEn2	Video Moment Retrieval					Highlight Detection	
						R1		mAP			>=Very Good	
						@0.5	@0.7	@0.5	@0.75	Avg.	mAP	HIT@1
(a)		✓	✓	✓	✓	$63.82_{\pm 0.66}$	$49.24_{\pm 0.77}$	$63.94_{\pm 0.11}$	$44.65_{\pm 0.28}$	$44.28_{\pm 0.26}$	$39.95_{\pm 0.27}$	$62.92_{\pm 1.02}$
(b)	✓		✓	✓	✓	$68.17_{\pm 0.88}$	$52.41_{\pm 0.96}$	$67.11_{\pm 0.73}$	$46.74_{\pm 0.96}$	$46.53_{\pm 0.50}$	$41.50_{\pm 0.25}$	$66.27_{\pm 0.95}$
(c)	✓	✓		✓	✓	$68.09_{\pm 0.65}$	$52.61_{\pm 0.84}$	$67.53_{\pm 0.49}$	$47.37_{\pm 0.42}$	$47.03_{\pm 0.17}$	$41.43_{\pm 0.15}$	$66.62_{\pm 0.43}$
(d)	✓	✓	✓		✓	$67.07_{\pm 1.54}$	$50.96_{\pm 2.17}$	$65.79_{\pm 1.72}$	$45.17_{\pm 2.10}$	$44.18_{\pm 1.91}$	$41.42_{\pm 0.09}$	$66.30_{\pm 0.29}$
(e)	✓	✓	✓	✓		$68.36_{\pm 0.31}$	$52.10_{\pm 0.36}$	$67.67_{\pm 0.17}$	$47.33_{\pm 0.65}$	$47.07_{\pm 0.36}$	$41.30_{\pm 0.24}$	$66.75_{\pm 0.92}$
(f)	✓	✓	✓	✓	✓	$\mathbf{69.01_{\pm 1.09}}$	$\mathbf{53.19_{\pm 0.38}}$	$\mathbf{68.43_{\pm 0.83}}$	$\mathbf{48.25_{\pm 0.59}}$	$\mathbf{47.93_{\pm 0.39}}$	$\mathbf{41.66_{\pm 0.15}}$	$\mathbf{66.80_{\pm 0.96}}$

Table 9: This table shows the necessery of each part in Convolutional Fuser. The model performs best when all parts are present. If remove any part of it, the model will perform worse. In this table, "V2TEx" means V2T Extractor, "T2VEn" means T2V Encoder, "TrEn1" means Transformer Encoder 1, "ConBl" means Convolutional Blocks, "TrEn2" means Transformer Encoder 2. The best result in each column is highlighted in **bold**.

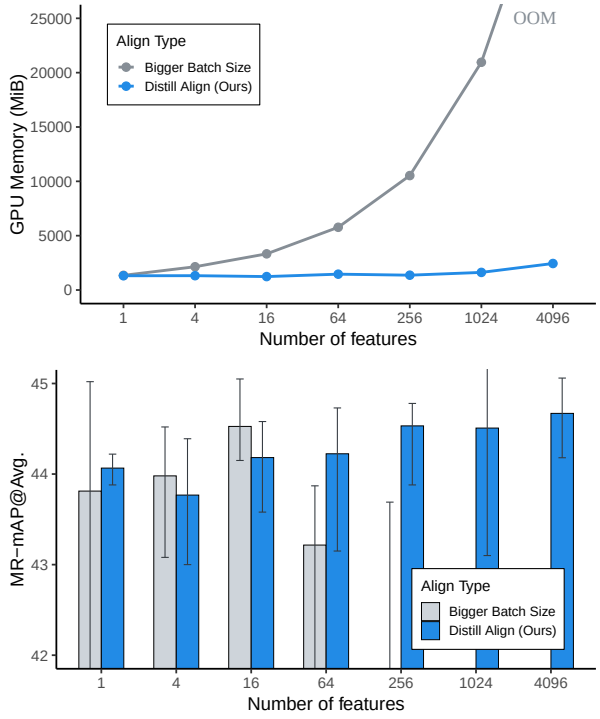

Figure 7: The Distill Align method can introduce more samples for comparative learning without taking up too much GPU memory. As the queue length increases, the model results get better. We visualized the GPU memory usage and the results on the QVHighlights dataset of the two methods when the number of features involved in contrastive learning increases. The experiment is conducted on an NVIDIA GeForce RTX 4090 with 24,564MiB GPU memory using TR-DETR (Sun et al. 2024) as the baseline. The x-axis represents the relative number of features involved in contrastive learning, where x-axis = 1 represents a batch size of 32 and a queue length of 0. "OOM" means Out of Memory. This figure shows the same set of experiments as Table 5.

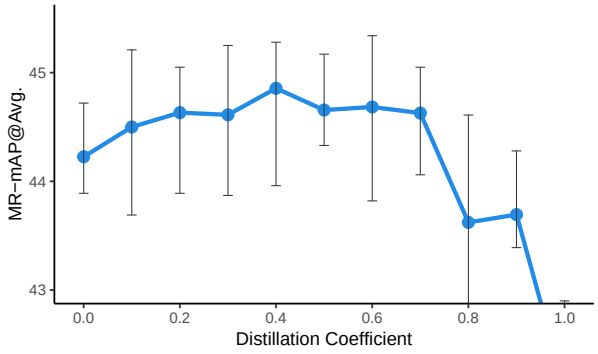

Figure 8: This figure shows the impact of distillation coefficient on multimodal alignment in Distill Align. This figure shows the same set of experiments as Table 6.

5 Conclusion

In this paper, we proposed a model LD-DETR for Video Moment Retrieval and Highlight Detection. We first proposed a plug-and-play method, *Distill Align*, which mitigates the impact of overlapping semantic information. Then, we introduced *Convolutional Fuser* which is more capable of capturing local information in multimodal features. Finally, we proposed a plug-and-play method, *Loop Decoder*, which allows multimodal information to be more adequately decoded without causing overfitting. The superiority and effectiveness of our approach have been demonstrated on four public datasets.

References

Anne Hendricks, L.; Wang, O.; Shechtman, E.; Sivic, J.; Darrell, T.; and Russell, B. 2017. Localizing moments in video with natural language. In *Proceedings of the IEEE international conference on computer vision*, 5803–5812.

Badamdorj, T.; Rochan, M.; Wang, Y.; and Cheng, L. 2021. Joint visual and audio learning for video highlight detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8127–8137.

Decoder Type	Decoder Size	Video Moment Retrieval					Highlight Detection	
		R1		mAP			>=Very Good	
		@0.5	@0.7	@0.5	@0.75	Avg.	mAP	HIT@1
Moment-DETR (Lei, Berg, and Bansal 2021)								
Bigger Decoder	1	$53.63_{\pm 1.59}$	$35.78_{\pm 0.25}$	$54.99_{\pm 1.33}$	$31.28_{\pm 1.07}$	$32.09_{\pm 1.01}$	$\mathbf{36.41_{\pm 0.40}}$	$\mathbf{56.92_{\pm 0.98}}$
	2	$54.71_{\pm 2.35}$	$35.84_{\pm 2.97}$	$55.80_{\pm 1.82}$	$31.39_{\pm 2.35}$	$32.50_{\pm 1.99}$	$36.17_{\pm 0.63}$	$56.63_{\pm 1.03}$
	3	$53.65_{\pm 0.73}$	$35.15_{\pm 1.28}$	$55.11_{\pm 0.68}$	$30.34_{\pm 0.63}$	$31.75_{\pm 0.73}$	$36.04_{\pm 0.37}$	$56.53_{\pm 0.98}$
	4	$52.72_{\pm 0.97}$	$35.14_{\pm 0.86}$	$54.65_{\pm 0.59}$	$30.64_{\pm 0.59}$	$31.59_{\pm 0.50}$	$35.78_{\pm 0.29}$	$55.48_{\pm 0.63}$
Loop Decoder	1	$53.63_{\pm 1.59}$	$35.78_{\pm 0.25}$	$54.99_{\pm 1.33}$	$31.28_{\pm 1.07}$	$32.09_{\pm 1.01}$	$\mathbf{36.41_{\pm 0.40}}$	$\mathbf{56.92_{\pm 0.98}}$
	2	$55.69_{\pm 0.46}$	$37.63_{\pm 0.55}$	$\mathbf{56.92_{\pm 0.45}}$	$32.93_{\pm 0.36}$	$\mathbf{33.86_{\pm 0.16}}$	$36.37_{\pm 0.41}$	$56.88_{\pm 1.23}$
	3	$54.93_{\pm 1.14}$	$37.88_{\pm 1.28}$	$56.19_{\pm 0.76}$	$33.09_{\pm 0.81}$	$33.66_{\pm 0.45}$	$35.83_{\pm 0.16}$	$55.30_{\pm 0.60}$
	4	$\mathbf{55.70_{\pm 0.90}}$	$\mathbf{38.24_{\pm 0.85}}$	$56.74_{\pm 0.93}$	$\mathbf{33.19_{\pm 0.70}}$	$33.85_{\pm 0.66}$	$36.10_{\pm 0.43}$	$56.43_{\pm 0.54}$
BM-DETR (Jung et al. 2023)								
Bigger Decoder	1	$60.90_{\pm 0.79}$	$44.10_{\pm 1.43}$	$60.91_{\pm 0.96}$	$39.03_{\pm 1.46}$	$38.93_{\pm 0.98}$	-	-
	2	$61.24_{\pm 1.33}$	$44.13_{\pm 1.42}$	$61.25_{\pm 0.88}$	$39.21_{\pm 1.06}$	$38.96_{\pm 1.03}$	-	-
	3	$60.72_{\pm 0.54}$	$44.14_{\pm 1.00}$	$61.14_{\pm 0.43}$	$39.74_{\pm 1.05}$	$39.18_{\pm 0.68}$	-	-
	4	$60.82_{\pm 0.89}$	$43.73_{\pm 0.78}$	$61.20_{\pm 0.84}$	$38.97_{\pm 0.80}$	$38.83_{\pm 0.81}$	-	-
Loop Decoder	1	$60.90_{\pm 0.79}$	$44.10_{\pm 1.43}$	$60.91_{\pm 0.96}$	$39.03_{\pm 1.46}$	$38.93_{\pm 0.98}$	-	-
	2	$61.01_{\pm 0.70}$	$\mathbf{44.80_{\pm 0.96}}$	$61.68_{\pm 0.96}$	$\mathbf{40.34_{\pm 0.93}}$	$39.77_{\pm 0.75}$	-	-
	3	$61.42_{\pm 1.20}$	$44.70_{\pm 0.95}$	$61.71_{\pm 1.06}$	$39.75_{\pm 0.27}$	$39.24_{\pm 0.63}$	-	-
	4	$\mathbf{61.62_{\pm 0.73}}$	$44.76_{\pm 1.48}$	$\mathbf{61.74_{\pm 0.47}}$	$40.14_{\pm 0.92}$	$\mathbf{39.81_{\pm 0.87}}$	-	-
QD-DETR (Moon et al. 2023b)								
Bigger Decoder	1	$61.98_{\pm 0.55}$	$47.30_{\pm 0.69}$	$62.03_{\pm 0.43}$	$41.96_{\pm 0.66}$	$41.42_{\pm 0.28}$	$38.92_{\pm 0.30}$	$62.05_{\pm 0.86}$
	2	$62.59_{\pm 0.70}$	$47.48_{\pm 0.79}$	$62.02_{\pm 0.75}$	$41.90_{\pm 0.63}$	$41.37_{\pm 0.54}$	$39.17_{\pm 0.21}$	$62.18_{\pm 0.67}$
	3	$61.73_{\pm 1.01}$	$45.38_{\pm 1.63}$	$60.45_{\pm 1.22}$	$39.99_{\pm 1.75}$	$39.57_{\pm 1.20}$	$\mathbf{39.24_{\pm 0.23}}$	$\mathbf{62.99_{\pm 0.74}}$
	4	$60.96_{\pm 0.43}$	$44.94_{\pm 1.03}$	$59.92_{\pm 0.40}$	$39.53_{\pm 0.76}$	$39.04_{\pm 0.77}$	$38.82_{\pm 0.27}$	$61.39_{\pm 0.93}$
Loop Decoder	1	$61.98_{\pm 0.55}$	$47.30_{\pm 0.69}$	$62.03_{\pm 0.43}$	$41.96_{\pm 0.66}$	$41.42_{\pm 0.28}$	$38.92_{\pm 0.30}$	$62.05_{\pm 0.86}$
	2	$\mathbf{62.80_{\pm 0.80}}$	$47.82_{\pm 0.53}$	$62.97_{\pm 0.49}$	$42.93_{\pm 0.61}$	$42.26_{\pm 0.37}$	$39.23_{\pm 0.13}$	$62.83_{\pm 0.85}$
	3	$62.49_{\pm 0.37}$	$47.57_{\pm 0.44}$	$\mathbf{63.12_{\pm 0.47}}$	$42.68_{\pm 0.36}$	$42.32_{\pm 0.32}$	$38.94_{\pm 0.19}$	$61.88_{\pm 0.43}$
	4	$62.45_{\pm 0.86}$	$\mathbf{47.95_{\pm 0.42}}$	$63.03_{\pm 0.41}$	$\mathbf{43.20_{\pm 0.08}}$	$\mathbf{42.48_{\pm 0.24}}$	$39.14_{\pm 0.15}$	$62.67_{\pm 0.54}$
CG-DETR (Moon et al. 2023a)								
Bigger Decoder	1	$65.92_{\pm 0.22}$	$50.44_{\pm 0.54}$	$65.50_{\pm 0.22}$	$45.62_{\pm 0.68}$	$44.76_{\pm 0.26}$	$40.34_{\pm 0.20}$	$65.12_{\pm 0.64}$
	2	$65.81_{\pm 1.23}$	$50.09_{\pm 0.54}$	$64.40_{\pm 0.31}$	$44.17_{\pm 0.51}$	$43.71_{\pm 0.37}$	$40.33_{\pm 0.25}$	$65.10_{\pm 0.40}$
	3	$65.30_{\pm 0.87}$	$49.22_{\pm 0.95}$	$63.50_{\pm 0.85}$	$43.66_{\pm 0.45}$	$42.85_{\pm 0.45}$	$\mathbf{40.45_{\pm 0.20}}$	$\mathbf{65.61_{\pm 0.50}}$
	4	$63.87_{\pm 2.51}$	$46.92_{\pm 4.35}$	$61.63_{\pm 3.07}$	$41.30_{\pm 3.54}$	$40.78_{\pm 3.12}$	$40.37_{\pm 0.11}$	$65.03_{\pm 0.57}$
Loop Decoder	1	$65.92_{\pm 0.22}$	$50.44_{\pm 0.54}$	$65.50_{\pm 0.22}$	$45.62_{\pm 0.68}$	$44.76_{\pm 0.26}$	$40.34_{\pm 0.20}$	$65.12_{\pm 0.64}$
	2	$65.56_{\pm 0.49}$	$50.58_{\pm 0.59}$	$65.24_{\pm 0.44}$	$45.29_{\pm 0.59}$	$44.61_{\pm 0.39}$	$40.10_{\pm 0.28}$	$64.95_{\pm 0.79}$
	3	$\mathbf{66.26_{\pm 0.64}}$	$\mathbf{51.72_{\pm 0.53}}$	$\mathbf{65.74_{\pm 0.39}}$	$45.66_{\pm 0.27}$	$45.05_{\pm 0.32}$	$40.39_{\pm 0.24}$	$65.53_{\pm 0.47}$
	4	$66.22_{\pm 0.50}$	$51.04_{\pm 0.36}$	$66.01_{\pm 0.51}$	$\mathbf{45.95_{\pm 0.52}}$	$\mathbf{45.23_{\pm 0.35}}$	$40.33_{\pm 0.13}$	$65.50_{\pm 0.91}$
TR-DETR (Sun et al. 2024)								
Bigger Decoder	1	$66.56_{\pm 1.06}$	$50.13_{\pm 0.89}$	$65.70_{\pm 0.79}$	$45.10_{\pm 0.78}$	$44.33_{\pm 0.51}$	$40.88_{\pm 0.19}$	$65.54_{\pm 0.45}$
	2	$66.13_{\pm 1.09}$	$49.54_{\pm 1.04}$	$64.02_{\pm 1.63}$	$43.86_{\pm 1.11}$	$43.35_{\pm 0.91}$	$40.61_{\pm 0.37}$	$65.10_{\pm 1.00}$
	3	$64.18_{\pm 1.22}$	$46.79_{\pm 1.25}$	$62.46_{\pm 0.96}$	$42.12_{\pm 0.91}$	$41.33_{\pm 0.85}$	$\mathbf{41.04_{\pm 0.16}}$	$65.28_{\pm 0.57}$
	4	$64.71_{\pm 0.94}$	$46.92_{\pm 1.35}$	$62.69_{\pm 0.71}$	$42.26_{\pm 1.25}$	$41.54_{\pm 0.86}$	$40.85_{\pm 0.10}$	$\mathbf{65.65_{\pm 0.88}}$
Loop Decoder	1	$66.56_{\pm 1.06}$	$50.13_{\pm 0.89}$	$65.70_{\pm 0.79}$	$45.10_{\pm 0.78}$	$44.33_{\pm 0.51}$	$40.88_{\pm 0.19}$	$65.54_{\pm 0.45}$
	2	$66.59_{\pm 0.63}$	$51.10_{\pm 0.40}$	$65.97_{\pm 0.20}$	$45.42_{\pm 0.20}$	$44.90_{\pm 0.11}$	$40.93_{\pm 0.29}$	$65.60_{\pm 0.81}$
	3	$\mathbf{67.12_{\pm 0.36}}$	$51.08_{\pm 0.47}$	$\mathbf{66.35_{\pm 0.46}}$	$\mathbf{45.58_{\pm 0.47}}$	$\mathbf{44.92_{\pm 0.33}}$	$40.78_{\pm 0.44}$	$65.37_{\pm 0.80}$
	4	$66.94_{\pm 0.72}$	$\mathbf{51.40_{\pm 0.51}}$	$66.04_{\pm 0.69}$	$45.22_{\pm 0.64}$	$44.90_{\pm 0.34}$	$40.85_{\pm 0.11}$	$65.63_{\pm 0.69}$

Table 10: This page has too little space to write on. See Table 11 for details.

Decoder Type	Decoder Size	Video Moment Retrieval					Highlight Detection	
		R1		mAP			>=Very Good	
		@0.5	@0.7	@0.5	@0.75	Avg.	mAP	HIT@1
UVCOM (Xiao et al. 2024)								
Bigger Decoder	1	$64.49_{\pm 1.10}$	$49.94_{\pm 0.72}$	$64.03_{\pm 1.01}$	$44.88_{\pm 0.53}$	$44.39_{\pm 0.61}$	$39.94_{\pm 0.18}$	$64.03_{\pm 0.25}$
	2	$64.37_{\pm 0.33}$	$48.73_{\pm 0.73}$	$62.85_{\pm 0.88}$	$43.52_{\pm 1.16}$	$42.73_{\pm 0.81}$	$39.97_{\pm 0.16}$	$63.42_{\pm 1.09}$
	3	$63.21_{\pm 1.36}$	$47.52_{\pm 1.81}$	$61.61_{\pm 1.19}$	$43.23_{\pm 1.35}$	$42.36_{\pm 1.24}$	$39.81_{\pm 0.13}$	$63.44_{\pm 0.15}$
	4	$62.68_{\pm 2.90}$	$46.17_{\pm 3.85}$	$60.74_{\pm 2.57}$	$41.40_{\pm 2.81}$	$40.99_{\pm 2.84}$	$39.90_{\pm 0.48}$	$63.72_{\pm 0.64}$
Loop Decoder	1	$64.49_{\pm 1.10}$	$49.94_{\pm 0.72}$	$64.03_{\pm 1.01}$	$44.88_{\pm 0.53}$	$44.39_{\pm 0.61}$	$39.94_{\pm 0.18}$	$64.03_{\pm 0.25}$
	2	$63.81_{\pm 0.37}$	$49.50_{\pm 0.60}$	$63.60_{\pm 0.38}$	$44.44_{\pm 0.37}$	$44.03_{\pm 0.24}$	$39.81_{\pm 0.10}$	$63.37_{\pm 0.38}$
	3	$64.55_{\pm 0.69}$	$50.04_{\pm 0.95}$	$64.11_{\pm 0.54}$	$45.41_{\pm 1.90}$	$44.26_{\pm 0.56}$	$39.94_{\pm 0.17}$	$63.94_{\pm 0.46}$
	4	$\mathbf{64.65_{\pm 0.61}}$	$\mathbf{50.33_{\pm 0.55}}$	$\mathbf{64.17_{\pm 0.46}}$	$44.54_{\pm 0.44}$	$44.18_{\pm 0.29}$	$39.76_{\pm 0.18}$	$63.69_{\pm 0.80}$
Loop Decoder w/o Slot Attention	1	$64.36_{\pm 0.44}$	$50.21_{\pm 1.26}$	$63.99_{\pm 0.27}$	$\mathbf{45.52_{\pm 0.68}}$	$\mathbf{44.72_{\pm 0.53}}$	$39.85_{\pm 0.21}$	$63.82_{\pm 1.05}$
	2	$64.36_{\pm 0.54}$	$50.06_{\pm 0.80}$	$63.85_{\pm 0.30}$	$45.15_{\pm 0.68}$	$44.43_{\pm 0.51}$	$39.88_{\pm 0.24}$	$\mathbf{64.19_{\pm 0.59}}$
	3	$64.00_{\pm 0.64}$	$49.30_{\pm 1.22}$	$63.73_{\pm 0.54}$	$44.33_{\pm 1.00}$	$43.93_{\pm 0.63}$	$39.79_{\pm 0.20}$	$63.54_{\pm 0.74}$
	4	$63.39_{\pm 0.61}$	$49.02_{\pm 0.50}$	$63.32_{\pm 0.31}$	$43.88_{\pm 0.52}$	$43.20_{\pm 0.35}$	$39.35_{\pm 0.18}$	$62.59_{\pm 0.09}$
Loop Decoder w/o Clip-Text Alignment	1	$63.96_{\pm 0.37}$	$49.73_{\pm 0.42}$	$63.65_{\pm 0.40}$	$44.55_{\pm 0.47}$	$44.12_{\pm 0.24}$	$39.82_{\pm 0.20}$	$63.63_{\pm 0.94}$
	2	$\mathbf{64.65_{\pm 0.53}}$	$50.09_{\pm 0.59}$	$\mathbf{64.17_{\pm 0.72}}$	$44.79_{\pm 0.87}$	$44.42_{\pm 0.61}$	$39.83_{\pm 0.17}$	$63.90_{\pm 0.41}$
	3	$64.62_{\pm 0.77}$	$50.10_{\pm 0.40}$	$64.12_{\pm 0.64}$	$44.74_{\pm 0.44}$	$44.25_{\pm 0.35}$	$\mathbf{39.98_{\pm 0.15}}$	$63.74_{\pm 0.36}$
	4	$63.85_{\pm 0.78}$	$49.55_{\pm 0.45}$	$63.60_{\pm 1.13}$	$44.87_{\pm 0.67}$	$44.15_{\pm 0.82}$	$35.72_{\pm 7.91}$	$63.54_{\pm 0.67}$

Table 11: Loop Decoder as a plug-and-play method can help multiple models achieve better results without the risk of overfitting like larger decoders. The table categorizes the models by baselines. For Loop Decoder, Decoder Size represents the number of Loop Decoder loops, and for Bigger Decoder, Decoder Size represents the multiple of the number of Decoder layers. Slot Attention and Clip-Text Alignment are two methods used in the UVCOM (Xiao et al. 2024) model. The best result in each category of features in each column is highlighted in **bold**. This table shows the same set of experiments as Figure 9.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Cai, S.; Zuo, W.; Davis, L. S.; and Zhang, L. 2018. Weakly-supervised video summarization using variational encoder-decoder and web prior. In *Proceedings of the European conference on computer vision (ECCV)*, 184–200.

Chen, G.; Huang, Y.; Xu, J.; Pei, B.; Chen, Z.; Li, Z.; Wang, J.; Li, K.; Lu, T.; and Wang, L. 2024. Video mamba suite: State space model as a versatile alternative for video understanding. *arXiv preprint arXiv:2403.09626*.

Chen, J.; Chen, X.; Ma, L.; Jie, Z.; and Chua, T.-S. 2018. Temporally grounding natural sentence in video. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, 162–171.

Chen, S.; and Jiang, Y.-G. 2019. Semantic proposal for activity localization in videos via sentence query. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 8199–8206.

Chen, Z.; Du, H.; Wu, Y.; Xu, T.; and Chen, E. 2020. Cross-modal video moment retrieval based on visual-textual relationship alignment. *Science in China Press: SCIENTIA SINICA Informationis*, 50(06): 862–876. (in Chinese).

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Escorcia, V.; Soldan, M.; Sivic, J.; Ghanem, B.; and Russell, B. 2019a. Finding moments in video collections using natural language. *arXiv preprint arXiv:1907.12763*.

Escorcia, V.; Soldan, M.; Sivic, J.; Ghanem, B.; and Russell, B. 2019b. Temporal localization of moments in video collections with natural language.

Feichtenhofer, C.; Fan, H.; Malik, J.; and He, K. 2019. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6202–6211.

Gao, J.; Sun, C.; Yang, Z.; and Nevatia, R. 2017. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE international conference on computer vision*, 5267–5275.

Gao, J.; Sun, X.; Xu, M.; Zhou, X.; and Ghanem, B. 2021a. Relation-aware video reading comprehension for temporal language grounding. *arXiv preprint arXiv:2110.05717*.

Gao, P.; Zheng, M.; Wang, X.; Dai, J.; and Li, H. 2021b. Fast convergence of detr with spatially modulated co-attention. In *Proceedings of the IEEE/CVF international conference on computer vision*, 3621–3630.

Gygli, M.; Song, Y.; and Cao, L. 2016. Video2gif: Automatic generation of animated gifs from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1001–1009.

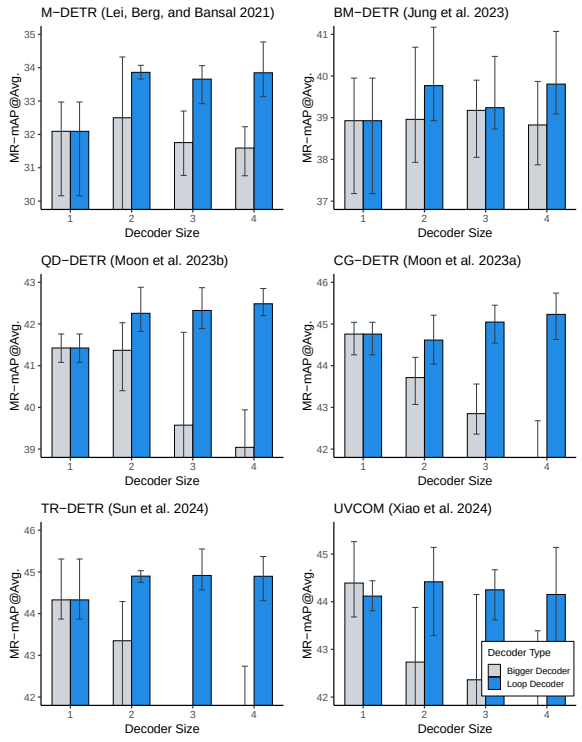

Figure 9: Loop Decoder as a plug-and-play method can help multiple models achieve better results without the risk of overfitting. For Loop Decoder, the x-axis represents the number of Loop Decoder loops. For Bigger Decoder, the x-axis represents the multiple of the number of Decoder layers. This figure shows the same set of experiments as Table 10 and Table 11.

Han, D.; Cheng, X.; Guo, N.; Ye, X.; Rainer, B.; and Priller, P. 2023. Momentum cross-modal contrastive learning for video moment retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9729–9738.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hoffmann, D. T.; Behrmann, N.; Gall, J.; Brox, T.; and Noroozi, M. 2022. Ranking info noise contrastive estimation: Boosting contrastive learning via ranked positives. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 897–905.

Hong, F.-T.; Huang, X.; Li, W.-H.; and Zheng, W.-S. 2020. Mini-net: Multiple instance ranking network for video highlight detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, 345–360. Springer.

Huang, C.; Wu, Y.-L.; Shuai, H.-H.; and Huang, C.-C. 2024. Semantic Fusion Augmentation and Semantic Boundary Detection: A Novel Approach to Multi-Target Video Moment Retrieval. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 6783–6792.

Jang, J.; Park, J.; Kim, J.; Kwon, H.; and Sohn, K. 2023. Knowing where to focus: Event-aware transformer for video grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 13846–13856.

Ji, W.; Liang, R.; Zheng, Z.; Zhang, W.; Zhang, S.; Li, J.; Li, M.; and Chua, T.-s. 2023. Are binary annotations sufficient? video moment retrieval via hierarchical uncertainty-based active learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 23013–23022.

Jiang, X.; Xu, X.; Shen, F.; Wang, G.; and Yang, Y. 2023. Efficient Weakly-Supervised Video Moment Retrieval without Multimodal Fusion. *Journal of Beijing University of Aeronautics and Astronautics*, 1–12. (in Chinese).

Jiang, Y.; Zhang, W.; Zhang, X.; Wei, X.; Chen, C. W.; and Li, Q. 2024. Prior Knowledge Integration via LLM Encoding and Pseudo Event Regulation for Video Moment Retrieval. *arXiv preprint arXiv:2407.15051*.

Jing, W.; Sun, A.; Zhang, H.; and Li, X. 2023. MS-DETR: Natural Language Video Localization with Sampling Moment-Moment Interaction. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1387–1400. Toronto, Canada: Association for Computational Linguistics.

Jung, M.; Jang, Y.; Choi, S.; Kim, J.; Kim, J.-H.; and Zhang, B.-T. 2023. Overcoming Weak Visual-Textual Alignment for Video Moment Retrieval. *arXiv preprint arXiv:2306.02728*.

Kong, Q.; Cao, Y.; Iqbal, T.; Wang, Y.; Wang, W.; and Plumbley, M. D. 2020. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28: 2880–2894.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Lee, P.; and Byun, H. 2023. BAM-DETR: Boundary-Aligned Moment Detection Transformer for Temporal Sentence Grounding in Videos. *arXiv preprint arXiv:2312.00083*.

Lei, J.; Berg, T. L.; and Bansal, M. 2021. Detecting moments and highlights in videos via natural language queries. *Advances in Neural Information Processing Systems*, 34: 11846–11858.

Lei, J.; Yu, L.; Berg, T. L.; and Bansal, M. 2020. Tvr: A large-scale dataset for video-subtitle moment retrieval. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, 447–463. Springer.

Li, J. 2024. *Research on Video Content Grounding based on Cross-modal Information Interaction*. Master's thesis, Sun Yat-sen University. (in Chinese).

Li, J.; Zhang, F.; Lin, S.; Zhou, F.; and Wang, R. 2023. Mim: Lightweight multi-modal interaction model for joint video moment retrieval and highlight detection. In *2023 IEEE International Conference on Multimedia and Expo (ICME)*, 1961–1966. IEEE.

Li, K.; Guo, D.; and Wang, M. 2021. Proposal-free video grounding with contextual pyramid network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 1902–1910.

Li, P.; Xie, C.-W.; Xie, H.; Zhao, L.; Zhang, L.; Zheng, Y.; Zhao, D.; and Zhang, Y. 2024. Momentdiff: Generative video moment retrieval from random to real. *Advances in neural information processing systems*, 36.

Lin, K. Q.; Zhang, P.; Chen, J.; Pramanick, S.; Gao, D.; Wang, A. J.; Yan, R.; and Shou, M. Z. 2023. Univtg: Towards unified video-language temporal grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2794–2804.

Liu, D.; Qu, X.; Dong, J.; Zhou, P.; Cheng, Y.; Wei, W.; Xu, Z.; and Xie, Y. 2021. Context-aware biaffine localizing network for temporal sentence grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11235–11244.

Liu, M.; Wang, X.; Nie, L.; Tian, Q.; Chen, B.; and Chua, T.-S. 2018. Cross-modal moment localization in videos. In *Proceedings of the 26th ACM international conference on Multimedia*, 843–851.

Liu, S.; Li, F.; Zhang, H.; Yang, X.; Qi, X.; Su, H.; Zhu, J.; and Zhang, L. 2022a. Dab-detr: Dynamic anchor boxes are better queries for detr. *arXiv preprint arXiv:2201.12329*.

Liu, W.; Mei, T.; Zhang, Y.; Che, C.; and Luo, J. 2015. Multi-task deep visual-semantic embedding for video thumbnail selection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3707–3715.

Liu, W.; Miao, B.; Cao, J.; Zhu, X.; Liu, B.; Nasim, M.; and Mian, A. 2024a. Context-Enhanced Video Moment Retrieval with Large Language Models. *arXiv preprint arXiv:2405.12540*.

Liu, Y.; He, J.; Li, W.; Kim, J.; Wei, D.; Pfister, H.; and Chen, C. W. 2024b. R^2-Tuning: Efficient Image-to-Video Transfer Learning for Video Temporal Grounding. *arXiv preprint arXiv:2404.00801*.

Liu, Y.; Li, S.; Wu, Y.; Chen, C.-W.; Shan, Y.; and Qie, X. 2022b. Umt: Unified multi-modal transformers for joint video moment retrieval and highlight detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3042–3051.

Liu, Z.; Li, J.; Xie, H.; Li, P.; Ge, J.; Liu, S.-A.; and Jin, G. 2024c. Towards balanced alignment: Modal-enhanced semantic modeling for video moment retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 3855–3863.

Loshchilov, I. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Ma, K.; Fang, H.; Zang, X.; Ban, C.; Zhou, L.; He, Z.; Li, Y.; Sun, H.; Feng, Z.; and Hou, X. 2024. Disentangle and denoise: Tackling context misalignment for video moment retrieval. *arXiv preprint arXiv:2408.07600*.

Ma, K.; Zang, X.; Feng, Z.; Fang, H.; Ban, C.; Wei, Y.; He, Z.; Li, Y.; and Sun, H. 2023. LLaViLo: Boosting Video Moment Retrieval via Adapter-Based Multimodal Modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2798–2803.

Ma, Y.; Yang, T.; Shan, Y.; and Li, X. 2022. Simvtp: Simple video text pre-training with masked autoencoders. *arXiv preprint arXiv:2212.03490*.

Meng, D.; Chen, X.; Fan, Z.; Zeng, G.; Li, H.; Yuan, Y.; Sun, L.; and Wang, J. 2021. Conditional detr for fast training convergence. In *Proceedings of the IEEE/CVF international conference on computer vision*, 3651–3660.

Miao, L.; Zhang, W.; and Xu, L. 2024. Pre-training model for video moment retrieval based on CLIP. *Application Research of Computers*, 1–8. (in Chinese).

Moon, W.; Hyun, S.; Lee, S.; and Heo, J.-P. 2023a. Correlation-guided Query-Dependency Calibration in Video Representation Learning for Temporal Grounding. *arXiv preprint arXiv:2311.08835*.

Moon, W.; Hyun, S.; Park, S.; Park, D.; and Heo, J.-P. 2023b. Query-dependent video representation for moment retrieval and highlight detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23023–23033.

Nan, G.; Qiao, R.; Xiao, Y.; Liu, J.; Leng, S.; Zhang, H.; and Lu, W. 2021. Interventional video grounding with dual contrastive learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2765–2775.

Panta, L.; Shrestha, P.; Sapkota, B.; Bhattarai, A.; Manandhar, S.; and Sah, A. K. 2024. Cross-modal Contrastive Learning with Asymmetric Co-attention Network for Video Moment Retrieval. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 607–614.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Regneri, M.; Rohrbach, M.; Wetzel, D.; Thater, S.; Schiele, B.; and Pinkal, M. 2013. Grounding action descriptions in videos. *Transactions of the Association for Computational Linguistics*, 1: 25–36.

Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; and Savarese, S. 2019. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 658–666.

Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J. 1986. Learning representations by back-propagating errors. *nature*, 323(6088): 533–536.

Soldan, M.; Xu, M.; Qu, S.; Tegner, J.; and Ghanem, B. 2021. Vlg-net: Video-language graph matching network for video grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3224–3234.

Song, Y.; Redi, M.; Vallmitjana, J.; and Jaimes, A. 2016. To click or not to click: Automatic selection of beautiful thumbnails from videos. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, 659–668.

Song, Y.; Vallmitjana, J.; Stent, A.; and Jaimes, A. 2015. Tvsum: Summarizing web videos using titles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5179–5187.

Sun, H.; Zhou, M.; Chen, W.; and Xie, W. 2024. TR-DETR: Task-Reciprocal Transformer for Joint Moment Retrieval and Highlight Detection. *arXiv preprint arXiv:2401.02309*.

Tian, Y.; Krishnan, D.; and Isola, P. 2020. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, 776–794. Springer.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, L.; Liu, D.; Puri, R.; and Metaxas, D. N. 2020. Learning trailer moments in full-length movies with co-contrastive attention. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, 300–316. Springer.

Wang, R.; Feng, J.; Zhang, F.; Luo, X.; and Luo, Y. 2024. Modality-aware Heterogeneous Graph for Joint Video Moment Retrieval and Highlight Detection. *IEEE Transactions on Circuits and Systems for Video Technology*.

Wang, T.; Zhang, J.; Zheng, F.; Jiang, W.; Cheng, R.; and Luo, P. 2023. Learning grounded vision-language representation for versatile understanding in untrimmed videos. *arXiv preprint arXiv:2303.06378*.

Wang, W.; Huang, Y.; and Wang, L. 2019. Language-driven temporal activity localization: A semantic matching reinforcement learning model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 334–343.

Wang, Y.; Zhang, X.; Yang, T.; and Sun, J. 2022a. Anchor detr: Query design for transformer-based detector. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, 2567–2575.

Wang, Z.; Wang, L.; Wu, T.; Li, T.; and Wu, G. 2022b. Negative sample matters: A renaissance of metric learning for temporal grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 2613–2623.

Wei, F.; Wang, B.; Ge, T.; Jiang, Y.; Li, W.; and Duan, L. 2022. Learning pixel-level distinctions for video highlight detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3073–3082.

Xiao, S.; Chen, L.; Zhang, S.; Ji, W.; Shao, J.; Ye, L.; and Xiao, J. 2021. Boundary proposal network for two-stage natural language video localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 2986–2994.

Xiao, Y.; Luo, Z.; Liu, Y.; Ma, Y.; Bian, H.; Ji, Y.; Yang, Y.; and Li, X. 2024. Bridging the gap: A unified video comprehension framework for moment retrieval and highlight detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18709–18719.

Xiong, B.; Kalantidis, Y.; Ghadiyaram, D.; and Grauman, K. 2019. Less is more: Learning highlight detection from video duration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1258–1267.

Xu, M.; Wang, H.; Ni, B.; Zhu, R.; Sun, Z.; and Wang, C. 2021. Cross-category video highlight detection via set-based learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7970–7979.

Xu, Y.; Sun, Y.; Zhai, B.; Jia, Y.; and Du, S. 2024. Mh-detr: Video moment and highlight detection with cross-modal transformer. In *2024 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Yan, S.; Xiong, X.; Nagrani, A.; Arnab, A.; Wang, Z.; Ge, W.; Ross, D.; and Schmid, C. 2023. Unloc: A unified framework for video localization tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 13623–13633.

Yang, J.; Liu, Y.; Song, L.; and Yan, X. 2022. Cross-modal Video Moment Retrieval Based on Enhancing Significant Features. *Journal of Electronics and Information Technology*, 44(12): 4395–4404. (in Chinese).

Yang, J.; Wei, P.; Li, H.; and Ren, Z. 2024. Task-Driven Exploration: Decoupling and Inter-Task Feedback for Joint Moment Retrieval and Highlight Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18308–18318.

Yao, T.; Mei, T.; and Rui, Y. 2016. Highlight detection with pairwise deep ranking for first-person video summarization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 982–990.

Yao, Z.; Ai, J.; Li, B.; and Zhang, C. 2021. Efficient detr: improving end-to-end object detector with dense prior. *arXiv preprint arXiv:2104.01318*.

Ye, Q.; Shen, X.; Gao, Y.; Wang, Z.; Bi, Q.; Li, P.; and Yang, G. 2021. Temporal cue guided video highlight detection with low-rank audio-visual fusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7950–7959.

Yu, S.; Cho, J.; Yadav, P.; and Bansal, M. 2024. Self-chained image-language model for video localization and question answering. *Advances in Neural Information Processing Systems*, 36.

Yu, Y.; Lee, S.; Na, J.; Kang, J.; and Kim, G. 2018. A deep ranking model for spatio-temporal highlight detection from a 360◦ video. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Yuan, L.; Tay, F. E. H.; Li, P.; and Feng, J. 2019a. Unsupervised video summarization with cycle-consistent adversarial LSTM networks. *IEEE Transactions on Multimedia*, 22(10): 2711–2722.

Yuan, Y.; Ma, L.; Wang, J.; Liu, W.; and Zhu, W. 2019b. Semantic conditioned dynamic modulation for temporal sentence grounding in videos. *Advances in Neural Information Processing Systems*, 32.

Yuan, Y.; Mei, T.; and Zhu, W. 2019. To find where you talk: Temporal sentence localization in video with attention based location regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 9159–9166.

Zeng, R.; Xu, H.; Huang, W.; Chen, P.; Tan, M.; and Gan, C. 2020. Dense regression network for video grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10287–10296.

Zhang, D.; Dai, X.; Wang, X.; Wang, Y.-F.; and Davis, L. S. 2019. Man: Moment alignment network for natural language moment retrieval via iterative graph adjustment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1247–1257.

Zhang, H.; Sun, A.; Jing, W.; and Zhou, J. T. 2020a. Span-based localizing network for natural language video localization. *arXiv preprint arXiv:2004.13931*.

Zhang, K.; Chao, W.-L.; Sha, F.; and Grauman, K. 2016. Video summarization with long short-term memory. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*, 766–782. Springer.

Zhang, M.; Yang, Y.; Chen, X.; Ji, Y.; Xu, X.; Li, J.; and Shen, H. T. 2021. Multi-stage aggregated transformer network for temporal language localization in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12669–12678.

Zhang, S.; Peng, H.; Fu, J.; and Luo, J. 2020b. Learning 2d temporal adjacent networks for moment localization with natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 12870–12877.

Zheng, M.; Gao, P.; Zhang, R.; Li, K.; Wang, X.; Li, H.; and Dong, H. 2020. End-to-end object detection with adaptive clustering transformer. *arXiv preprint arXiv:2011.09315*.

Zhou, S.; Zhang, F.; Wang, R.; Zhou, F.; and Su, Z. 2024. Subtask Prior-driven Optimized Mechanism on Joint Video Moment Retrieval and Highlight Detection. *IEEE Transactions on Circuits and Systems for Video Technology*.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2020. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*.